FORM PTO-1390 (Modified)
(REV 11-98)

U.S. DEPARTMENT OF COMMERCE PATENT AND TRADEMARK OFFICE

TRANSMITTAL LETTER TO THE UNITED STATES DESIGNATED/ELECTED OFFICE (DO/EO/US) CONCERNING A FILING UNDER 35 U.S.C. 371

ATTORNEY'S DOCKET NUMBER
T2147-907163

U.S. APPLICATION NO. (IF KNOWN, SEE 37 CFR
09/831878

INTERNATIONAL APPLICATION NO.	INTERNATIONAL FILING DATE	PRIORITY DATE CLAIMED
PCT/FR00/02469	**7 September 2000**	**16 September 1999**

TITLE OF INVENTION
RELAY FOR ACCESSING A SERVER NETWORK, TRANSPARENT TO A CLIENT NETWORK

APPLICANT(S) FOR DO/EO/US
Jean-Yves DUJONC and Rene MARTIN

Applicant herewith submits to the United States Designated/Elected Office (DO/EO/US) the following items and other information:

1. ☒ This is a **FIRST** submission of items concerning a filing under 35 U.S.C. 371.
2. ☐ This is a **SECOND** or **SUBSEQUENT** submission of items concerning a filing under 35 U.S.C. 371.
3. ☒ This is an express request to begin national examination procedures (35 U.S.C. 371(f)) at any time rather than delay examination until the expiration of the applicable time limit set in 35 U.S.C. 371(b) and PCT Articles 22 and 39(1).
4. ☒ A proper Demand for International Preliminary Examination was made by the 19th month from the earliest claimed priority date.
5. ☒ A copy of the International Application as filed (35 U.S.C. 371 (c) (2))
   a. ☒ is transmitted herewith (required only if not transmitted by the International Bureau).
   b. ☐ has been transmitted by the International Bureau.
   c. ☐ is not required, as the application was filed in the United States Receiving Office (RO/US).
6. ☒ A translation of the International Application into English (35 U.S.C. 371(c)(2)).
7. ☒ A copy of the International Search Report (PCT/ISA/210).
8. ☐ Amendments to the claims of the International Application under PCT Article 19 (35 U.S.C. 371 (c)(3))
   a. ☐ are transmitted herewith (required only if not transmitted by the International Bureau).
   b. ☐ have been transmitted by the International Bureau.
   c. ☐ have not been made; however, the time limit for making such amendments has NOT expired.
   d. ☐ have not been made and will not be made.
9. ☐ A translation of the amendments to the claims under PCT Article 19 (35 U.S.C. 371(c)(3)).
10. ☒ An oath or declaration of the inventor(s) (35 U.S.C. 371 (c)(4)).
11. ☐ A copy of the International Preliminary Examination Report (PCT/IPEA/409).
12. ☐ A translation of the annexes to the International Preliminary Examination Report under PCT Article 36 (35 U.S.C. 371 (c)(5)).

Items 13 to 20 below concern document(s) or information included:

13. ☒ An Information Disclosure Statement under 37 CFR 1.97 and 1.98.
14. ☒ An assignment document for recording. A separate cover sheet in compliance with 37 CFR 3.28 and 3.31 is included.
15. ☒ A **FIRST** preliminary amendment.
16. ☐ A **SECOND** or **SUBSEQUENT** preliminary amendment.
17. ☒ A substitute specification.
18. ☒ A change of power of attorney and/or address letter.
19. ☐ Certificate of Mailing by Express Mail
20. ☒ Other items or information:

Verification of Translator
Formal Drawings (2)
PCT Forms: PCT/IB/301; PCT/IB/304; PCT/IB/308; PCT/RO/101; Demande
Copies of prior art cited in International Search Report

U.S. APPLICATION NO. (IF KNOWN, SEE 37 CFR	INTERNATIONAL APPLICATION NO.	ATTORNEY'S DOCKET NUMBER
09/831878	PCT/FR00/02469	T2147-907163

21. The following fees are submitted:.	CALCULATIONS	PTO USE ONLY
BASIC NATIONAL FEE (37 CFR 1.492 (a) (1) - (5)) :		
☐ Neither international preliminary examination fee (37 CFR 1.482) nor international search fee (37 CFR 1.445(a)(2) paid to USPTO and International Search Report not prepared by the EPO or JPO **$1,000.00**		
☒ International preliminary examination fee (37 CFR 1.482) not paid to USPTO but Internation Search Report prepared by the EPO or JPO **$860.00**		
☐ International preliminary examination fee (37 CFR 1.482) not paid to USPTO but international search fee (37 CFR 1.445(a)(2)) paid to USPTO **$710.00**		
☐ International preliminary examination fee paid to USPTO (37 CFR 1.482) but all claims did not satisfy provisions of PCT Article 33(1)-(4) **$690.00**		
☐ International preliminary examination fee paid to USPTO (37 CFR 1.482) and all claims satisfied provisions of PCT Article 33(1)-(4) **$100.00**		
ENTER APPROPRIATE BASIC FEE AMOUNT =	$860.00	
Surcharge of **$130.00** for furnishing the oath or declaration later than ☐ 20 ☐ 30 months from the earliest claimed priority date (37 CFR 1.492 (e)).	$0.00	

CLAIMS	NUMBER FILED	NUMBER EXTRA	RATE		
Total claims	9 - 20 =	0	x $18.00	$0.00	
Independent claims	3 - 3 =	0	x $80.00	$0.00	
Multiple Dependent Claims **(check if applicable).**			☐	$0.00	
	TOTAL OF ABOVE CALCULATIONS =			$860.00	

Reduction of 1/2 for filing by small entity, if applicable. Verified Small Entity Statement must also be filed (Note 37 CFR 1.9, 1.27, 1.28) **(check if applicable).**	☐	$0.00	
SUBTOTAL =		$860.00	
Processing fee of **$130.00** for furnishing the English translation later than ☐ 20 ☐ 30 months from the earliest claimed priority date (37 CFR 1.492 (f)). +		$0.00	
TOTAL NATIONAL FEE =		$860.00	
Fee for recording the enclosed assignment (37 CFR 1.21(h)). The assignment must be accompanied by an appropriate cover sheet (37 CFR 3.28, 3.31) **(check if applicable).**	☒	$40.00	
TOTAL FEES ENCLOSED =		$900.00	
		Amount to be: refunded	$
		charged	$

☒ A check in the amount of **$900.00** to cover the above fees is enclosed.

☐ Please charge my Deposit Account No. in the amount of to cover the above fees. A duplicate copy of this sheet is enclosed.

☒ The Commissioner is hereby authorized to charge any fees which may be required, or credit any overpayment to Deposit Account No. **501165** A duplicate copy of this sheet is enclosed.

NOTE: Where an appropriate time limit under 37 CFR 1.494 or 1.495 has not been met, a petition to revive (37 CFR 1.137(a) or (b)) must be filed and granted to restore the application to pending status.

SEND ALL CORRESPONDENCE TO:

Edward J. Kondracki
MILES & STOCKBRIDGE P.C.
Suite 500
1751 Pinnacle Drive
McLean, VA 22102-3833

SIGNATURE

Edward J. Kondracki
NAME

20,064
REGISTRATION NUMBER

May 16, 2001
DATE

T2147-907163-US3782/JMD/PG(PCT)

UNITED STATES DESIGNATED/ELECTED OFFICE (D.O./E.O./US)

Applicant:	Jean-Yves DUJONC & Rene MARTIN
International Application No.:	PCT/FR 00/02469
International Filing Date:	7 September 2000
U.S. Serial No.:	
U.S. Filing Date:	May 16, 2001
For:	RELAY FOR ACCESSING A SERVER NETWORK, TRANPARENT TO A CLIENT NETWORK

McLean, Virginia
May 16, 2001

PRELIMINARY AMENDMENT

Assistant Commissioner of Patents
Washington, D.C. 20231

Sir:

Please substitute the attached specification, claims and abstract for the specification, claims and abstract in their entirety.

The substitute specification, claims and abstract are submitted in lieu of making multiple grammatical changes and revision of form necessary to the translation of the French text. An attempt has been made to adhere as closely as possible to the original language and no new matter has been added.

Prompt examination is earnestly solicited.

Respectfully submitted,

MILES & STOCKBRIDGE P.C.

By: Edward J. Kondracki

Edward J. Kondracki
Registration No. 20,604

1751 Pinnacle Drive – Suite 500
McLean, VA 22102-3833
Tel.: 703/903-9000
Fax: 703/610-8686

"CLEAN" COPY OF APPLICATION
Serial No.: ________, filed 5/16/2001
Our Ref.: T2147-907163

RELAY FOR ACCESSING A SERVER NETWORK, TRANSPARENT TO A CLIENT NETWORK

CROSS REFERENCE TO RELATED APPLICATIONS

The subject matter of this application is related to application Serial No.__________, filed ___________, Attorney Docket No.: T2147-907162, in the names of Nadine FABIANO, Bernard MAINGUENAUD and Rene MARTIN, entitled "METHOD FOR REDUCING CONGESTION IN A NETWORK" and corresponding to French Application No. 99 11592 and PCT Application No. PCT/FR 00/02470, incorporated herein in its entirety.

BACKGROUND OF THE INVENTION

1. Field of the Invention

The technical field to which the invention relates is that of computer networks. Computer networks make it possible to run distributed applications in remote machines linked to the same network or to different networks interconnected by means of interconnection machines.

2. Description of Related Art

A transaction between remote machines is initiated by a client application, which sends a request message to a server application in a standby state. The client application places itself in a wait state for a response message to its request message. Upon receiving the request message, the server application generates a response message that it sends to the client application. A network layer allows each message to be conveyed in the form of a datagram, from the machine hosting the sending application to the machine hosting the receiving application. A transport layer allows the message to be conveyed between the sending application and the network layer, then between the network layer and the receiving application, for example from a client application to a server application. An application layer handles the execution of the application in its own environment.

When the machines are not physically linked to the same network, routing protocols of the network layer route the datagrams from the sending machine to an interconnection machine, and from the interconnection machine to the receiving machine, using internetwork protocol addresses, such as for example IP addresses. When passing through the interconnection machine, the datagrams remain at the network layer level. The network between the client machine and the interconnection machine is called the client network. The network between the server machine and the interconnection machine is called the server network.

The technical field to which the invention particularly relates involves an interconnection machine for hosting a relay application, or proxy. A relay application is useful for performing operations on the messages exchanged between the client network and the server network. However, datagrams addressed to the final receiving machine are naturally not sent up to the application layer of the relay machine.

According to the known prior art, the sending application addresses its messages to the relay application of the relay machine instead of addressing them directly to the final receiving application, and indicates in its messages to the relay application the final application to which its messages are to be sent so that the relay application can reroute them by means of the operations it applies to them. This is what happens, for example in an Internet browser, in which it is possible to declare, for a given client application, the address of the relay machine for the network layer and the port number of the relay application for the transport layer, so that the browser encapsulates the address of the server machine and the port number of the final destination application in a datagram addressed to the relay application. However, this makes it necessary to know the relay application through which the messages must pass in order to configure the client machine accordingly. The resulting lack of flexibility, while acceptable for a limited number of applications, is unsatisfactory for a large number of different applications.

The document RFC1928, available on the internet at the address http://www.pmg.lcs.mit.edu/cgi-bin/rfc/view?1928, describes the protocol "SOCKS v5," wherein the port number conventionally used is 1080. Just as for the solution known as "TCP protocol tunneling in web proxy servers," it is necessary to establish a first connection to the relay application, followed by a second connection of the relay machine to the final machine.

SUMMARY OF THE INVENTION

In order to eliminate the drawbacks mentioned above, the object of the present invention is to allow a client application to simply establish a connection to a server application the way it would when not using the services of a relay application, so that the use of the services of the relay application is transparent for the client application.

A first embodiment of the present invention is a relay machine linked to a client network by means of a first physical interface and linked to a server network by means of a second physical interface, characterized in that at least one internetwork protocol address of a server machine linked to the server network is associated with the first physical interface, and in that the relay machine comprises a first relay application for receiving datagrams addressed to the server machine from the client network and for sending to the server network datagrams addressed to the server machine.

Thus, when a datagram arrives in the first physical interface with the internetwork protocol address of the server machine as its destination address, the relay machine is recognized by its network layer as being the destination machine for the datagram. The network layer of the relay machine then sends the datagram up to the application layer of the relay machine by simply following the established protocol. When it receives this datagram, the relay application can process it, after which it may or may not retransmit it to the server machine. This is completely transparent for the client application.

In an alternative embodiment of the present invention, a relay machine is linked to a client network by means of a first physical interface and linked to a server network by means of a second physical interface, characterized in that at least one internetwork protocol address of a server machine linked to the server network is associated with a third physical interface, distinct from the first physical interface and from the second physical interface, and in that the relay machine comprises a first relay application for receiving datagrams addressed to the server machine from the client network and for sending to the server network datagrams addressed to the server machine.

In this case, the protocol of the network layer does not require the destination address to be assigned to the first physical interface that receives the datagram, but instead, to any physical interface of the relay machine, so that the destination address is sent up to the application layer of the relay machine.

When the relay machine already has a base address in the client network, useful, for example, for routing protocols, the server machine address is associated with the first physical interface as a synonym address of the base address of the relay machine in the client network.

The present invention includes a method for processing, by means of a relay application running in a relay machine between a client network and a server network, datagrams sent through the client network by a client application, addressed to a server machine having an address in the server network, characterized in that the method includes a first step that associates the address in the server network with a physical interface of the relay machine that is not linked to the server network, so that the relay application receives the datagrams.

This offers the advantage of making it unnecessary to configure or inform the client application in order for the relay application to be able to process the datagrams. In essence, the client application continues to send its datagrams using the address of the server machine. When the datagram arrives in the first physical interface of the relay machine, the network protocol ensures that the datagram is naturally sent up to the application layer of the relay machine, thus allowing the relay application to receive it.

When it is necessary to route the datagrams transmitted from the client network to the server network through the relay machine, the method is characterized in that the first step is preceded by a second step for routing the datagrams transmitted through the client network, addressed to the server machine, to the relay machine. This is the case, for example, when there is more than one relay machine between the client network and the server network.

BRIEF DESCRIPTION OF THE DRAWINGS

Other advantages and details of the implementation of the invention will emerge from the following description in reference to the figures, in which:

- Fig. 1 represents an exemplary relay machine with two physical interfaces according to the present invention;

- Fig. 2 represents an exemplary datagram, according to the present invention;

- Fig. 3 represents an exemplary relay machine with three physical interfaces, according to the present invention.

DESCRIPTION OF THE PREFERRED EMBODIMENTS

In Fig. 1, according to the present invention, represents server machines 1, 2 and client machines 11, 12. The machines 1, 2, 11 are linked to a server network 3 by means of respective physical interfaces 7, 8, 17. A client machine 12 is linked to a client network 13 by means of a physical interface 18. The networks 3 and 13 are physically separate. A relay machine 4 is linked to the server network 3 by means of a physical interface 14 and to the network 13 by means of a physical interface 19.

The applications 5, 6, 15, 16 running in the machines 1, 2, 11, 12 communicate with one another through a transport layer CT using a protocol in the connectionless mode such as UDP, or in the connected mode such as TCP. The transport layer CT supervises a network layer CR using a protocol such as IP.

In the network layer CR, the machine 1 is recognized by means of an address @S1, the machine 2 is recognized by means of an address @S2, and the machine 11 is recognized by means of an address @C1. In a known way, each of the addresses @S1, @S2 and @C1 has a network field with a common value that identifies the network 3, and a machine field with a distinct value that identifies each machine linked to the network 3. The machine 12 is recognized by means of an address @C2 with a network field value that identifies the network 13 and a machine field value that identifies the machine 12 in the network 13. The machine 4 is recognized by means of an address @P1 with a network field value that identifies the network 13 and a machine field value that identifies the machine 4 in the network 13, and by means of an address @P2 with a network field value that identifies the network 3 and a machine field value that identifies the machine 4 in the network 3.

The machines communicate with one another by means of messages that flow through the networks in the form of datagrams. Fig. 2 presents an exemplary datagram according to the present invention. This datagram, constituted by a frame of successive bits, is structured in three successive fields. A first field marked DR is dedicated to the protocol of the network layer. A second field marked DT is dedicated to the protocol of the transport layer that supervises the network layer. A third field marked DA is dedicated to an application layer that supervises the transport layer. In the case of a request on the web, for example, the field DR contains the source and destination IP addresses, the field DT contains the source and destination TCP port numbers, and the field DA contains HTTP data.

For example, if a client application 15 running in the client machine 11 issues a request to access a file processed by a server application 5 located in the server machine 1, the application 15 transmits its request to the layer CT of the machine 11, which writes the request into the field DA, and writes into the field DT a service port number for the application 15 and a service port number for the application 5. The layer CT of the machine 11 transmits the fields DT and DA to the layer CR of the machine 11, which writes into the field DR the address @C1 of the machine 11 and the address @S1 of the machine 1. The layer CR then transmits through the interface 17 the datagram thus constituted, which arrives through the interface 7 of the machine 1. The layer CR of the machine 1 recognizes from the address @S1 that the datagram is to be sent to the upper layers of the machine 1, and retransmits the fields DT and DA to the layer CT of the machine 1. Using the service port number for the application 5, the layer CT retransmits the field DA to the application 5, which processes the request.

If an application 16 running in the client machine 12 issues a request to access a file processed by the application 5 located in the server machine 1, the application 16 transmits its request to the layer CT of the machine 12, which writes it into the field DA and which writes into the field DT a service port number for the application 16 and a service port number for the application 5. The layer CT of the machine 12 transmits the fields DT and DA to the layer CR of the machine 12, which writes into the field DR the address @C2 of the machine 12 and the address @S1 of the machine 1. The layer CR then transmits the datagram thus constituted to the interface 18 that arrives through the interface 19 of the machine 4, which operates as a router between the networks 13 and 3.

According to the present invention, the layer CR of the machine 4 recognizes that the datagram is not to be sent to the upper layers of the machine 4 because @S1 is not a destination address of the machine 4. The layer CR of the machine 4 then searches in routing tables for a line containing a value identical to the network field of the address @S1. The line thus found indicates the interface 14 as being the one for accessing the network 3. The layer CR of the machine 4 therefore retransmits the datagram to the network 3 through the interface 14 so that the datagram arrives through the interface 7 of the machine 1. The layer CR of the machine 1 recognizes from the address @S1 that the datagram is to be sent to the upper layers of the machine 1 and retransmits the fields DT and DA to the layer CT of the machine 1.

Using the service port number for the application 5, the layer CT retransmits the field DA to the application 5,'which processes the request.

With the device according to the invention, the machine 4 comprises an application 22 that plays the role of a relay, or proxy server, for requests issuing from the network 13. The application 22 offers several advantages, such as, for example, it can control access to the machines 1, 2, 11 linked to the server network 3, and it can save responses to previous requests in a cache in order to restore these responses for new requests without requiring these new requests to be routed to the server machine 1, 2.

The layer CR includes several addresses which are associated with the physical interface 19, including the usual address @P1 and the address @S1 of the server machine 1 linked to the network 3. It is also possible to associate the address @S2 of the server machine 2 with the physical interface 19. As made clear by the description below, unlike the prior art in which it is the client network that determines the utilization of the services of the relay application 22, in the arrangement of the present invention it is the server network that determines this utilization. For example, access to the server 1, is accomplished by associating the address @S1 with the physical interface 19.

The application 22 comprises an input port 9 with the same number as the input port of the application 5, and an output port 10 to which it can assign a number, in order to handle any request messages addressed to the application 5.

As a result of this particular device, the machine 12 does not need to know that it is establishing an intermediate connection with the machine 4. If an application 16 running in the client machine 12 issues a request addressed to the application 5 located in the server machine 1, the address @S1 is then recognized in the network 13 as being the address of the machine 4.

In order to issue a request addressed to the application 5, the application 16 sends a datagram Q through the network 13 that contains the addresses @S1 and @C2 in the field CR, the port numbers of the applications 5 an 16 in the transport field, and the final information addressed to the application 5 in the field CA.

When the datagram Q is received through the physical interface 19 of the machine 4, the network layer CR of the machine 4 recognizes the destination address @S1 in the field DR as being an address that belongs to it, and therefore sends the datagram up to the transport layer CT of the machine 4. The transport layer CT recognizes the destination number in the field DT as being the number of the port 9 of the application 22, to which it then transmits the content of the datagram Q.

The application 22 then processes the content of the field DA of the datagram Q. The processing of the datagram Q by the application 22 consists, for example, of verifying access rights, and checking to see if the machine 4 already contains a response to the request in its cache in order to decide whether or not to communicate the datagram Q to the server application 5.

When, in order to process the request message received from the client application 16, the application 22 needs to send a request message to the application 5, the application 22 communicates the following data to the transport layer CT of the machine 4: the content of the request to be entered into the field DA, the input port number of the application 5, an output port number of the application 22 for handling the response to the request, and the internetwork protocol address @S1 of the machine 1. These data are transmitted to the network layer CR of the machine 4. Upon receiving these data, the network layer CR of the machine 4 searches in its routing tables for the network through which to send a datagram, based on the network field of the address @S1. In the example described here, the network field of the address @S1 corresponds to the network 3 to which the machine 1 is linked, and the layer CR sends to the physical interface 14 a datagram containing in the field DR the destination address @S1and the source address @P2 associated with the physical interface 14. In the server network 3, the datagram conventionally reaches the machine 1 and the server application 5 in the machine 1.

The response received from the application 5 through the interface 14 is sent to the application 22 by the network layer because the address @P2 is an address of the machine 4, and is then transmitted to application 22 by the transport layer CT because the port number previously identified for the response is the one assigned to the port 10 by the application 22. Using an internal request and response handling mechanism, the application 22 associates the response with the outgoing port number received from the application 16. In order to retransmit the response to the application 16, the application 22 communicates the following data to the transport layer CT of the machine 4: the content of the response to be entered into the field DA, the output port number of the application 16, the input port number of the application 22 which is identical to the input port number of the application 5 for handling the response to the request, the destination internetwork protocol address @C2 of the machine 12 and the source internetwork protocol address @S1 of the machine 1. These data are transmitted to the network layer CR of the machine 4 by the transport layer. Upon receiving these data, the network layer CR of the machine 4 searches in its routing tables for the network to which to send a datagram, based on the network field of the address @C2. In the example described here, the network field of the address @C2 corresponding to the network 13 to which the machine 12 is linked, the layer CR sends to the physical interface 19 a datagram that contains, in the field DR, the destination address @P2 and the source address @S1 associated with the physical interface 19. In the client network 13, the datagram conventionally reaches the machine 12 and the client application 16 in the machine 12.

Thus, the application 16 in the machine 12 receives a response that is returned by the application 5 in the machine 1 without having to pass through the application 22; this occurs in a way that is transparent for the client application 16.

Referring to Fig. 3, the address @S1 is associated with a physical interface 20 that is different both from the interface 14 as in the preceding case, and from the interface 19 as in this particular case.

When a datagram is sent through the network 13 with the address @S1, the routing protocol of the network layer CR of the machine 4 detects it in the interface 19 with which the address @P1 is associated. Since the address @S1 associated with the physical interface 20 is an address of the machine 4, the datagram is sent up to the application layer CA of the machine 4.

A relay application 21 processes the request message obtained from the datagram received, just like the preceding relay application 22. In order to send the response message to the application 12, the relay application 22 has a specific driver to a virtual network to which the physical interface 20 is linked.

The case in which the IP address @S1 is associated with the interface 19 is particularly advantageous for making the invention easy to use. In the simple example that follows, the application 16 executes a Telnet function as a client application, and the application 22 executes a telnetd function as a server application of the application 16 and a Telnet function as a client of the application 5. The application 5 executes a telnetd function as a server of the application 22. Telnet and telnetd are known functions that use TCP/IP to connect a terminal of a client machine in which the Telnet function is executed to a server machine in which the telnetd function is executed.

In order to keep track of the machine in which the commands are executed, each machine runs on a different operating system. The client machine 12 runs on an AIX (registered trademark) version 4.1 system, and has the IP address @C1 = 129.182.51.58. The relay machine 4 runs on an AIX version 4.2 system and has the IP addresses @P1 = 129.182.51.21 and @P2 = 192.90.249.22. The server machine 12 runs on a (proprietary) DNS-E system and has the IP address @S1 = 192.90.249.124. The network 13 is accessible in a known way at an IP address @R1 = 129.182.50 with a mask @M1 = 255.255.254.0.

In the client machine 12, the command

```
route add –host 192.90.249.124 129.182.51.21
```

means that in order to reach the server machine 1 with the address @S1, the datagrams sent pass through the relay machine with the address @P1.

In the server machine 1, the command

```
route add –net 129.182.50 192.90.249.22 –netmask
255.255.254.0
```

means that in order to reach any machine of the network 13 with the address @R1, the datagrams sent pass through the relay machine with the address @P2.

In the client machine 12, the command

```
Telnet 192.90.249.124
```

activates the Telnet application in order to reach the server machine 1 with the address @S1. At this stage, the only machine recognized through the IP address @S1 is the server machine 1. The IP layer of the machine 4 routes the datagrams sent by the IP layer of the machine 12 to the IP layer of the server machine 1. The IP layer of the machine 1, recognizing the address @S1, sends the application field of the datagrams to the telnetd application of the machine 1. In return, the telnetd application of the machine 1 sends the machine 12 the message:

```
Trying...
Connected to 192.90.249.124.
Escape character is '^]'.
$$ 0000 *DNS-E V3U1.000 P1.001 P2.019 P3.010*IMA:BX77SIM
1998/10/21 17:23*
```

The display of this message on the terminal of the machine 12 shows that it is in the DNS system environment, which means that the machine 1 has been reached directly. The relay machine 4 was not passed through in order to perform the IP routing.

In the client machine 12, the command

```
Telnet 129.182.51.21
```

activates the Telnet application in order to reach the relay machine 4 with the address @P1. The IP layer of the machine 4, recognizing the address @P1, sends the application field of the datagrams to the telnetd application of the machine 4. In return, the telnetd application of the machine 4 sends the machine 12 the message

```
Trying...
Connected to 129.182.51.21.
Escape character is '^]'.
Telnet (thirteen)
AIX Version 4
© Copyrights by IBM and by others 1982, 1996.
Login:
```

The display of this message on the terminal of the machine 12 shows that it is in the AIX system environment, which means that the machine 4 has been reached. This makes it possible to generate commands from the terminal of the machine 12 that are executed in the machine 4.

In the machine 4, the interface 19 being named en1, the command:

```
ifconfig en1 192.90.249.124 alias
```

defines the address @S1 as an additional address associated with the interface 19. The machine 4 runs no risk of being confused with the machine 1 in the network 13 by the IP layer, since it is physically separate from the network 3. Likewise, the command:

```
ifconfig en1 192.90.249.125 alias
```

would define the address @S2 as an additional address associated with the interface 19.

Referring again to the machine 12, the command:

```
Telnet 192.90.249.124
```

activates the Telnet application with an effect that is different than the one described above. The message displayed on the terminal of the machine 12 is:

```
Trying...
Connected to 129.182.51.21.
Escape character is '^]'.
```

```
Telnet (thirteen)
AIX Version 4
© Copyrights by IBM and by others 1982, 1996.
Login:
```

The display of this message on the terminal of the machine 12 shows that the latter is in the AIX system environment of the machine 4. Despite having requested a connection to the telnetd application of the server machine 1 using the address @S1, the command has established a connection with the telnetd application of the machine 4. This is explained by the fact that the IP layer of the machine 4 recognizes the address @S1 as a destination address belonging to the machine 4, without taking into account the routing through the network 3. Thus, the IP layer of the machine 4 sends the application field of the datagrams received through the interface 19 to the telnetd application of the machine 4.

At present, in the machine 4, the command:

```
Telnet 192.90.249.124
```

activates the Telnet application in order to reach the server machine 1 with the address @S1. At this stage, the only machine recognized by the IP address @S1 from the interface 14 is the server machine 1. The IP layer of the machine 1, recognizing the address @S1, sends the application field of the datagrams up to the telnetd application of the machine 1. In return, the telnetd application of the machine 1 sends to the Telnet application of the machine 4 the message:

```
Trying...
Connected to 192.90.249.124.
Escape character is '^]'.
$$ 0000 *DNS-E V3U1.000 P1.001 P2.019 P3.010*IMA:BX77SIM
1998/10/21 17:23*
```

This message is retransmitted by the telnetd application of the machine 4 to the Telnet application of the machine 12. The display of this message on the terminal of the machine 12 shows that it is in the DNS system environment, i.e., that the machine 1 has been reached. However, the application field of the datagrams is sent up to the application layer of the relay machine 4 in a way that is transparent for the machine 12.

The method explained above in terms of a manual operation can be implemented by means of a program executed by the application layer of the machine 4.

The datagrams addressed to the machine 1, which pass through the IP layer of the machine 4, are sent up to the application layer of the machine 4 because the address @S1 is associated with a physical interface of the machine 4. In order to avoid conflicts in the network 3 with the machine 1, it is preferable not to associate the address @S1 with the interface 14. Referring to Fig. 3, it is possible to associate the address @S1 with a physical interface other than the interface 19, for example a physical interface 20.

One example of a particular operation by the application 22 described here offers a particular advantage. If encryption keys are associated with the address @S1 in order to encrypt the requests received from and the responses sent to the machine 12, the decryption of the requests and the encryption of the responses can be handled by the machine 4. The decrypted data can flow through the server network 3 without any risk. Thus, the encryption and decryption resources can be centralized in the machine 4, leaving a maximum number of resources available in the machine 1 for its server functions. The application 22 is also responsible for re-encrypting the responses prior to sending them through the network 13.

SUMMARY

It should be clear to those skilled in the art that the present invention allows for embodiments in many other specific forms without going beyond the scope of application of the invention as claimed. Consequently, the present embodiments should be considered as examples which can be modified within the range defined by the true spirit and scope of the invention as set forth in the attached claims to which resort should be made for a full and complete understanding of the full scope of the invention.

CLAIMS

1. A relay machine (4) linked to a client network (13) by means of a first physical interface (19) and linked to a server network (3) distinct from the relay machine (4) by means of a second physical interface (14), the relay machine comprising at least one internetwork protocol address (@S1, @S2) of a server machine (1, 2) linked to the server network (3), said protocol address being associated with the first physical interface (19); and a first relay application (22) for receiving datagrams addressed to the server machine (1, 2) from the client network (13) and for sending to the server network (3) datagrams addressed to the server machine (1, 2).

2. A relay machine (4) linked to a client network (13) by means of a first physical interface (19) and linked to a server network (3) distinct from the relay machine (4) by means of a second physical interface (14), the relay machine comprising at least one internetwork protocol address (@S1, @S2) of a server machine (1, 2) linked to the server network (3), said protocol address being associated with a third physical interface (20), distinct from the first physical interface (19) and from the second physical interface (14)[,]; and [in that it comprises] a first relay application (22) for receiving datagrams addressed to the server machine (1, 2) from the client network (13) and for sending to the server network (3) datagrams addressed to the server machine (1, 2).

3. The relay machine (4) according to claim 1, wherein said address (@S1, @S2) is associated with the first physical interface (19), said protocol address as an address synonymous with a base address (@P1) of the machine (4) in the network (13).

4. A method for processing, by means of at least one relay application (22) running in a relay machine (4) between a client network (13) and a server network (3), datagrams sent through the client network (13) by a client application (16) to a server machine (1) with a protocol address (@S1) in the server network (3), distinct from the relay machine (4), the step comprising: associating said address (@S1) with a physical interface (19, 20) of the relay machine (4) that is not linked to the server

network (3), so that the relay application (22) receives said datagrams without the need to configure or inform said client application (16) in order to receive said datagrams.

5. The method according to claim 4, wherein the step of associating is preceded by a step of routing the datagrams transmitted through the client network (13), addressed to the server machine (1), to the relay machine (4).

6. The relay machine (4) according to claim 1, the application (22) includes encryption keys and further comprising transmitting encrypted messages received from the network (13) in decrypted fashion inside the network (3).

7. The relay machine (4) according to claim 2, the application (22) includes encryption keys and further comprising transmitting encrypted messages received from the network (13) in decrypted fashion inside the network (3).

8. The relay machine (4) according to claim 1, the application (22) includes encryption keys and further comprising transmitting unencrypted messages received from the network (3) in encrypted fashion inside the network (13).

9. The relay machine (4) according to claim 1, the application (22) includes encryption keys and further comprising transmitting unencrypted messages received from the network (3) in encrypted fashion inside the network (13).

ABSTRACT

RELAY FOR ACCESSING A SERVER NETWORK, TRANSPARENT TO A CLIENT NETWORK

A relay machine (4) linked to a client network (13) by means of a first physical interface (19) and linked to a server network (3) by means of a second physical interface (14). The relay machine (4) comprises a first relay application (22) for receiving datagrams addressed to the server machine (1, 2) from the network (13) and for sending to the network (3) datagrams addressed to the server machine (1, 2). An internetwork protocol address (@S1, @S2) of a server machine (1, 2) linked to the server network (3) is associated with the first physical interface (19) so that the datagrams sent up to the application level in the relay machine are available to the relay application in a way that is transparent to the client network (13).

[redacted]

AMENDED VERSION OF APPLICATION
~~Se~~rial No.:_________, filed 5/16/2001
(w/brackets & underlining)T2147-907163

RELAY FOR ACCESSING A SERVER NETWORK, TRANSPARENT TO A CLIENT NETWORK

CROSS REFERENCE TO RELATED APPLICATIONS

The subject matter of this application is related to application Serial No.__________, filed __________, Attorney Docket No.: T2147-907162, in the names of Nadine FABIANO, Bernard MAINGUENAUD and Rene MARTIN, entitled "METHOD FOR REDUCING CONGESTION IN A NETWORK" and corresponding to French Application No. 99 11592 and PCT Application No. PCT/FR 00/02470, incorporated herein in its entirety.

BACKGROUND OF THE INVENTION

1. Field of the Invention

The technical field to which the invention relates is that of computer networks. Computer networks make it possible to run distributed applications in remote machines linked to the same network or to different networks interconnected by means of interconnection machines.

2. Description of Related Art

A transaction between remote machines is initiated by a client application, which sends a request message to a server application in a standby state. The client application places itself in a wait state for a response message to its request message. Upon receiving the request message, the server application generates a response message that it sends to the client application. A network layer allows each message to be conveyed in the form of a datagram, from the machine hosting the sending application to the machine hosting the receiving application. A transport layer allows the message to be conveyed between the sending application and the network layer, then between the network layer and the receiving application, for example from a client application to a server application. An application layer handles the execution of the application in its own environment.

When the machines are not physically linked to the same network, routing protocols of the network layer route the datagrams from the sending machine to an interconnection machine, and from the interconnection machine to the receiving machine, using internetwork protocol addresses, such as for example IP addresses. When passing through the interconnection machine, the datagrams remain at the network layer level. The network between the client machine and the interconnection machine is called the client network. The network between the server machine and the interconnection machine is called the server network.

The technical field to which the invention particularly relates involves an interconnection machine for hosting a relay application, or proxy. A relay application is useful for performing operations on the messages exchanged between the client network and the server network. However, datagrams addressed to the final receiving machine are naturally not sent up to the application layer of the relay machine.

According to the known prior art, the sending application addresses its messages to the relay application of the relay machine instead of addressing them directly to the final receiving application, and indicates in its messages to the relay application the final application to which its messages are to be sent so that the relay application can reroute them by means of the operations it applies to them. This is what happens, for example in an Internet browser, in which it is possible to declare, for a given client application, the address of the relay machine for the network layer and the port number of the relay application for the transport layer, so that the browser encapsulates the address of the server machine and the port number of the final destination application in a datagram addressed to the relay application. However, this makes it necessary to know the relay application through which the messages must pass in order to configure the client machine accordingly. The resulting lack of flexibility, while acceptable for a limited number of applications, is unsatisfactory for a large number of different applications.

The document RFC1928, available on the internet at the address http://www.pmg.lcs.mit.edu/cgi-bin/rfc/view?1928, describes the protocol "SOCKS v5," wherein the port number conventionally used is 1080. Just as for the solution known as "TCP protocol tunneling in web proxy servers," it is necessary to establish a first connection to the relay application, followed by a second connection of the relay machine to the final machine.

SUMMARY OF THE INVENTION

In order to eliminate the drawbacks mentioned above, the object of the present invention is to allow a client application to simply establish a connection to a server application the way it would when not using the services of a relay application, so that the use of the services of the relay application is transparent for the client application.

A first [subject] embodiment of the present invention is a relay machine linked to a client network by means of a first physical interface and linked to a server network by means of a second physical interface, characterized in that at least one internetwork protocol address of a server machine linked to the server network is associated with the first physical interface, and in that [it] the relay machine comprises a first relay application for receiving datagrams addressed to the server machine from the client network and for sending to the server network datagrams addressed to the server machine.

Thus, when a datagram arrives in the first physical interface with the internetwork protocol address of the server machine as its destination address, the relay machine is recognized by its network layer as being the destination machine for the datagram. The network layer of the relay machine then sends the datagram up to the application layer of the relay machine by simply following the established protocol. When it receives this datagram, the relay application can process it, after which it may or may not retransmit it to the server machine. This is completely transparent for the client application.

[The subject of a variant] In an alternative embodiment of the present invention, [is] a relay machine is linked to a client network by means of a first physical interface and linked to a server network by means of a second physical interface, characterized in that at least one internetwork protocol address of a server machine linked to the server network is associated with a third physical interface, distinct from the first physical interface and from the second physical interface, and in that [it] the relay machine comprises a first relay application for receiving datagrams addressed to the server machine from the client network and for sending to the server network datagrams addressed to the server machine.

In this case, the protocol of the network layer does not require the destination address to be assigned to the first physical interface that receives the datagram, but instead, to any physical interface of the relay machine, so that [it] the destination address is sent up to the application layer of the relay machine.

When the relay machine already has a base address in the client network, useful, for example, for routing protocols, [said] the server machine address is associated with the first physical interface as a synonym address of the base address of the relay machine in the client network.

[A second subject of the] The present invention [is] includes a method for processing, by means of a relay application running in a relay machine between a client network and a server network, datagrams sent through the client network by a client application, addressed to a server machine having an address in the server network, characterized in that [it comprises] the method includes a first step that associates [said] the address in the server network with a physical interface of the relay machine that is not linked to the server network, so that the relay application receives [said] the datagrams.

This offers the advantage of making it unnecessary to configure or inform [said] the client application in order for the relay application to be able to process the datagrams. In essence, the client application continues to send its datagrams using the address of the server machine. When the datagram arrives in the first physical interface of the relay machine, the network protocol ensures that the datagram is naturally sent up to the application layer of the relay machine, thus allowing the relay application to receive it.

When it is necessary to route the datagrams transmitted from the client network to the server network through the relay machine, the method is characterized in that the first step is preceded by a second step for routing the datagrams transmitted through the client network, addressed to the server machine, to the relay machine. This is the case, for example, when there is more than one relay machine between the client network and the server network.

BRIEF DESCRIPTION OF THE DRAWINGS

Other advantages and details of the implementation of the invention will emerge from the following description in reference to the figures, in which:

- Fig. 1 represents an exemplary relay machine with two physical interfaces according to the present invention;

- Fig. 2 represents an exemplary datagram according to the present invention;

- Fig. 3 represents an exemplary relay machine with three physical interfaces according to the present invention.

DESCRIPTION OF THE PREFERRED EMBODIMENTS

In Fig. 1, according to the present invention, represents server machines 1, 2 and client machines 11, 12. The machines 1, 2, 11 are linked to a server network 3 by means of respective physical interfaces 7, 8, 17. A client machine 12 is linked to a client network 13 by means of a physical interface 18. The networks 3 and 13 are physically separate. A relay machine 4 is linked to the server network 3 by means of a physical interface 14 and to the network 13 by means of a physical interface 19.

The applications 5, 6, 15, 16 running in the machines 1, 2, 11, 12 communicate with one another through a transport layer CT using a protocol in the connectionless mode such as UDP, or in the connected mode such as TCP. The transport layer CT supervises a network layer CR using a protocol such as IP.

In the network layer CR, the machine 1 is recognized by means of an address @S1, the machine 2 is recognized by means of an address @S2, and the machine 11 is recognized by means of an address @C1. In a known way, each of the addresses @S1, @S2 and @C1 has a network field with a common value that identifies the network 3, and a machine field with a distinct value that identifies each machine linked to the network 3. The machine 12 is recognized by means of an address @C2 with a network field value that identifies the network 13 and a machine field value that identifies the machine 12 in the network 13. The machine 4 is recognized by means of an address @P1 with a network field value that identifies the network 13 and a machine field value that identifies the machine 4 in the network 13, and by means of an address @P2 with a network field value that identifies the network 3 and a machine field value that identifies the machine 4 in the network 3.

The machines communicate with one another by means of messages that flow through the networks in the form of datagrams. Fig. 2 presents an exemplary datagram according to the present invention. This datagram, constituted by a frame of successive bits, is [essentially] structured in three successive fields. A first field marked DR is dedicated to the protocol of the network layer. A second field marked DT is dedicated to the protocol of the transport layer that supervises the network layer. A third field marked DA is dedicated to an application layer that supervises the transport layer. In the case of a request on the web, for example, the field DR contains the source and destination IP addresses, the field DT contains the source and destination TCP port numbers, and the field DA contains HTTP data.

For example, if a client application 15 running in the client machine 11 issues a request to access a file processed by a server application 5 located in the server machine 1, the application [5] 15 transmits its request to the layer CT of the machine 11, which writes the request into the field DA, and writes into the field DT a service port number for the application 15 and a service port number for the application 5. The layer CT of the machine 11 transmits the fields DT and DA to the layer CR of the machine 11, which writes into the field DR the address @C1 of the machine 11 and the address @S1 of the machine 1. The layer CR then transmits through the interface 17 the datagram thus constituted, which arrives through the interface 7 of the machine 1. The layer CR of the machine 1 recognizes from the address @S1 that the datagram is to be sent to the upper layers of the machine 1, and retransmits the fields DT and DA to the layer CT of the machine 1. Using the service port number for the application 5, the layer CT retransmits the field DA to the application 5, which processes the request.

If an application 16 running in the client machine 12 issues a request to access a file processed by the application 5 located in the server machine 1, the application 16 transmits its request to the layer CT of the machine 12, which writes it into the field DA and which writes into the field DT a service port number for the application 16 and a service port number for the application 5. The layer CT of the machine 12 transmits the fields DT and DA to the layer CR of the machine 12, which writes into the field DR the address @C2 of the machine 12 and the address @S1 of the machine 1. The layer CR then transmits the datagram thus constituted to the interface 18 that arrives through the interface 19 of the machine 4, [declared as] which operates as a router between the networks 13 and 3.

[Without the device according] According to the present invention, [@S1 not being a destination address of the machine 4,] the layer CR of the machine 4 recognizes that the datagram is not to be sent to the upper layers of the machine 4 because @S1 is not a destination address of the machine 4. The layer CR of the machine 4 then searches in routing tables for a line containing a value identical to the network field of the address @S1. The line thus found indicates the interface 14 as being the one for accessing the network 3. The layer CR of the machine 4 therefore retransmits the datagram to the network 3 through the interface 14 so that the datagram arrives through the interface 7 of the machine 1. The layer CR of the machine 1 recognizes from the address @S1 that the datagram is to be sent to the upper layers of the machine 1 and retransmits the fields DT and DA to the layer CT of the machine 1. Using the service port number for the application 5, the layer CT retransmits the field DA to the application 5, which processes the request.

With the device according to the invention, the machine 4 comprises an application 22 that plays the role of a relay, or proxy server, for requests issuing from the network 13. The application 22 offers several advantages[;] , such as, for example, it can control access to the machines 1, 2, 11 linked to the server network 3, and it can save responses to previous requests in a cache in order to restore these responses for new requests without requiring these new requests to be routed to the server machine 1, 2.

[Several addresses of the] The layer CR includes several addresses which are associated with the physical interface 19, including the usual address @P1 and the address @S1 of the server machine 1 linked to the network 3. It is also possible to associate the address @S2 of the server machine 2 with the physical interface 19. As made clear by the description below, unlike the prior art in which it is the client network that determines the utilization of the services of the relay application 22, in [this case] the arrangement of the present invention it is the server network that determines this utilization[,]. [for] For example, [for accessing] access to the server 1, is accomplished by associating the address @S1 with the physical interface 19.

The application 22 comprises an input port 9 with the same number as the input port of the application 5, and an output port 10 to which it can assign a number, in order to handle any request messages addressed to the application 5.

As a result of this particular device, the machine 12 does not need to know that it is establishing an intermediate connection with the machine 4. If an application 16 running in the client machine 12 issues a request addressed to the application 5 located in the server machine 1, the address @S1 is then recognized in the network 13 as being the address of the machine 4.

In order to issue a request addressed to the application 5, the application 16 sends a datagram Q through the network 13 that contains the addresses @S1 and @C2 in the field CR, the port numbers of the applications 5 an 16 in the transport field, and the final information addressed to the application 5 in the field CA.

When the datagram Q is received through the physical interface 19 of the machine 4, the network layer CR of the machine 4 recognizes the destination address @S1 in the field DR as being an address that belongs to it, and therefore sends the datagram up to the transport layer CT of the machine 4. The transport layer CT recognizes the destination number in the field DT as being the number of the port 9 of the application 22, to which it then transmits the content of the datagram Q.

The application 22 then processes the content of the field DA of the datagram Q. The processing of the datagram Q by the application 22 consists, for example, of verifying access rights, and checking to see if the machine 4 already contains a response to the request in its cache in order to decide whether or not to communicate the datagram Q to the server application 5.

When, in order to process the request message received from the client application 16, the application 22 needs to send a request message to the application 5, the application 22 communicates the following data to the transport layer CT of the machine 4: the content of the request to be entered into the field DA, the input port number of the application 5, an output port number of the application 22 for handling the response to the request, and the internetwork protocol address @S1 of the machine 1. These data are transmitted to the network layer CR of the machine 4. Upon receiving these data, the network layer CR of the machine 4 searches in its routing tables for the network through which to send a datagram, based on the network field of the address @S1. In the example described here, the network field of the address @S1 [corresponding] corresponds to the network 3 to which the machine 1 is linked, and the layer CR sends to the physical interface 14 a datagram containing in the field DR the destination address @S1and the source address @P2 associated with the physical interface 14. In the server network 3, the datagram conventionally reaches the machine 1 and the server application 5 in the machine 1.

The response received from the application 5 through the interface 14 is sent to the application 22 by the network layer because the address @P2 is an address of the machine 4, and is then transmitted to application 22 by the transport layer CT because the port number previously identified for the response is the one assigned to the port 10 by the application 22. Using an internal request and response handling mechanism, the application 22 associates the response with the outgoing port number received from the application 16. In order to retransmit the response to the application 16, the application 22 communicates the following data to the transport layer CT of the machine 4: the content of the response to be entered into the field DA, the output port number of the application 16, the input port number of the application 22 which is identical to the input port number of the application 5 for handling the response to the request, the destination internetwork protocol address @C2 of the machine 12 and the source internetwork protocol address @S1 of the machine 1. These data are transmitted to the network layer CR of the machine 4 by the transport layer. Upon receiving these data, the network layer CR of the machine 4 searches in its routing tables for the network to which to send a datagram, based on the network field of the address @C2. In the example described here, the network field of the address @C2 corresponding to the network 13 to which the machine 12 is linked, the layer CR sends to the physical interface 19 a datagram that contains, in the field DR, the destination address @P2 and the source address @S1 associated with the physical interface 19. In the client network 13, the datagram conventionally reaches the machine 12 and the client application 16 in the machine 12.

Thus, the application 16 in the machine 12 receives a response that is returned by the application 5 in the machine 1 without having to pass through the application 22; this occurs in a way that is transparent for the client application 16.

Referring to Fig. 3, the address @S1 is associated with a physical interface 20 that is different both from the interface 14 as in the preceding case, and from the interface 19 as in this particular case.

When a datagram is sent through the network 13 with the address @S1, the routing protocol of the network layer CR of the machine 4 detects it in the interface 19 with which the address @P1 is associated. Since the address @S1 associated with the physical interface 20 is an address of the machine 4, the datagram is sent up to the application layer CA of the machine 4.

A relay application 21 processes the request message obtained from the datagram received, just like the preceding relay application 22. In order to send the response message to the application 12, the relay application 22 has a specific driver to a virtual network to which the physical interface 20 is linked.

The case in which the IP address @S1 is associated with the interface 19 is particularly advantageous for making the invention easy to use. In the simple example that follows, the application 16 executes a Telnet function as a client application, and the application 22 executes a telnetd function as a server application of the application 16 and a Telnet function as a client of the application 5. The application 5 executes a telnetd function as a server of the application 22. Telnet and telnetd are known functions that use TCP/IP to connect a terminal of a client machine in which the

Telnet function is executed to a server machine in which the telnetd function is executed.

In order to keep track of the machine in which the commands are executed, each machine runs on a different operating system. The client machine 12 runs on an AIX (registered trademark) version 4.1 system, and has the IP address @C1 = 129.182.51.58. The relay machine 4 runs on an AIX version 4.2 system and has the IP addresses @P1 = 129.182.51.21 and @P2 = 192.90.249.22. The server machine 12 runs on a (proprietary) DNS-E system and has the IP address @S1 = 192.90.249.124. The network 13 is accessible in a known way at an IP address @R1 = 129.182.50 with a mask @M1 = 255.255.254.0.

In the client machine 12, the command

```
route add –host 192.90.249.124 129.182.51.21
```

means that in order to reach the server machine 1 with the address @S1, the datagrams sent pass through the relay machine with the address @P1.

In the server machine 1, the command

```
route add –net 129.182.50 192.90.249.22 –netmask
255.255.254.0
```

means that in order to reach any machine of the network 13 with the address @R1, the datagrams sent pass through the relay machine with the address @P2.

In the client machine 12, the command

```
Telnet 192.90.249.124
```

activates the Telnet application in order to reach the server machine 1 with the address @S1. At this stage, the only machine recognized through the IP address @S1 is the server machine 1. The IP layer of the machine 4 routes the datagrams sent by the IP layer of the machine 12 to the IP layer of the server machine 1. The IP layer of the machine 1, recognizing the address @S1, sends the application field of the datagrams to the telnetd application of the machine 1. In return, the telnetd application of the machine 1 sends the machine 12 the message:

```
Trying...
Connected to 192.90.249.124.
Escape character is '^]'.
$$ 0000 *DNS-E V3U1.000 P1.001 P2.019 P3.010*IMA:BX77SIM
1998/10/21 17:23*
```

The display of this message on the terminal of the machine 12 shows that it is in the DNS system environment, which means that the machine 1 has been reached directly. The relay machine 4 was not passed through in order to perform the IP routing.

In the client machine 12, the command

```
Telnet 129.182.51.21
```

activates the Telnet application in order to reach the relay machine 4 with the address @P1. The IP layer of the machine 4, recognizing the address @P1, sends the application field of the datagrams to the telnetd application of the machine 4. In return, the telnetd application of the machine 4 sends the machine 12 the message

```
Trying...
Connected to 129.182.51.21.
Escape character is '^]'.
Telnet (thirteen)
AIX Version 4
© Copyrights by IBM and by others 1982, 1996.
Login:
```

The display of this message on the terminal of the machine 12 shows that it is in the AIX system environment, which means that the machine 4 has been reached. This makes it possible to generate commands from the terminal of the machine 12 that are executed in the machine 4.

In the machine 4, the interface 19 being named en1, the command:

```
ifconfig en1 192.90.249.124 alias
```

defines the address @S1 as an additional address associated with the interface 19. The machine 4 runs no risk of being confused with the machine 1 in the network 13 by the IP layer, since it is physically separate from the network 3. Likewise, the command:

```
ifconfig en1 192.90.249.125 alias
```

would define the address @S2 as an additional address associated with the interface 19.

Referring again to the machine 12, the command:

```
Telnet 192.90.249.124
```

activates the Telnet application with an effect that is different than the one described above. The message displayed on the terminal of the machine 12 is:

```
Trying...
```

```
Connected to 129.182.51.21.
Escape character is '^]'.
Telnet (thirteen)
AIX Version 4
© Copyrights by IBM and by others 1982, 1996.
Login:
```

The display of this message on the terminal of the machine 12 shows that the latter is in the AIX system environment of the machine 4. Despite having requested a connection to the telnetd application of the server machine 1 using the address @S1, the command has established a connection with the telnetd application of the machine 4. This is explained by the fact that the IP layer of the machine 4 recognizes the address @S1 as a destination address belonging to the machine 4, without taking into account the routing through the network 3. Thus, the IP layer of the machine 4 sends the application field of the datagrams received through the interface 19 to the telnetd application of the machine 4.

At present, in the machine 4, the command:

```
Telnet 192.90.249.124
```

activates the Telnet application in order to reach the server machine 1 with the address @S1. At this stage, the only machine recognized by the IP address @S1 from the interface 14 is the server machine 1. The IP layer of the machine 1, recognizing the address @S1, sends the application field of the datagrams up to the telnetd application of the machine 1. In return, the telnetd application of the machine 1 sends to the Telnet application of the machine 4 the message:

```
Trying...
Connected to 192.90.249.124.
Escape character is '^]'.
$$ 0000 *DNS-E V3U1.000 P1.001 P2.019 P3.010*IMA:BX77SIM
1998/10/21 17:23*
```

This message is retransmitted by the telnetd application of the machine 4 to the Telnet application of the machine 12. The display of this message on the terminal of the machine 12 shows that it is in the DNS system environment, i.e., that the machine 1 has been reached. However, the application field of the datagrams is sent up to the application layer of the relay machine 4 in a way that is transparent for the machine 12.

The method explained above in terms of a manual operation can be implemented by means of a program executed by the application layer of the machine 4.

The datagrams addressed to the machine 1, which pass through the IP layer of the machine 4, are sent up to the application layer of the machine 4 because the address @S1 is associated with a physical interface of the machine 4. In order to avoid conflicts in the network 3 with the machine 1, it is preferable not to associate the address @S1 with the interface 14. Referring to Fig. 3, it is possible to associate the address @S1 with a physical interface other than the interface 19, for example a physical interface 20.

One example of a particular operation by the application 22 described here offers a particular advantage. If encryption keys are associated with the address @S1 in order to encrypt the requests received from and the responses sent to the machine 12, the decryption of the requests and the encryption of the responses can be handled by the machine 4. The decrypted data can flow through the server network 3 without any risk. Thus, the encryption and decryption resources can be centralized in the machine 4, leaving a maximum number of resources available in the machine 1 for its server functions. The application 22 is also responsible for re-encrypting the responses prior to sending them through the network 13.

SUMMARY

It should be clear to those skilled in the art that the present invention allows for embodiments in many other specific forms without going beyond the scope of application of the invention as claimed. Consequently, the present embodiments should be considered as examples which can be modified within the range defined by the true spirit and scope of the invention as set forth in the attached claims to which resort should be made for a full and complete understanding of the full scope of the invention.

CLAIMS

1. A [Relay] relay machine (4) linked to a client network (13) by means of a first physical interface (19) and linked to a server network (3) distinct from the relay machine (4) by means of a second physical interface (14), [characterized in that] the relay machine comprising at least one internetwork protocol address (@S1, @S2) of a server machine (1, 2) linked to the server network (3), [distinct from the relay machine (4), is] said protocol address being associated with the first physical interface (19)[,]; and [in that it comprises] a first relay application (22) for receiving datagrams addressed to the server machine (1, 2) from the client network (13) and for sending to the server network (3) datagrams addressed to the server machine (1, 2).

2. A [Relay] relay machine (4) linked to a client network (13) by means of a first physical interface (19) and linked to a server network (3) distinct from the relay machine (4) by means of a second physical interface (14), [characterized in that] the relay machine comprising at least one internetwork protocol address (@S1, @S2) of a server machine (1, 2) linked to the server network (3), [distinct from the relay machine (4), is] said protocol address being associated with a third physical interface (20), distinct from the first physical interface (19) and from the second physical interface (14)[,]; and [in that it comprises] a first relay application (22) for receiving datagrams addressed to the server machine (1, 2) from the client network (13) and for sending to the server network (3) datagrams addressed to the server machine (1, 2).

3. The [Relay] relay machine (4) according to claim 1, [characterized in that] wherein said address (@S1, @S2) is associated with the first physical interface (19), said protocol address as an address synonymous with a base address (@P1) of the machine (4) in the network (13).

4. A [Method] method for processing, by means of at least one relay application (22) running in a relay machine (4) between a client network (13) and a server network (3), datagrams sent through the client network (13) by a client application (16) to a server machine (1) with [the] a protocol address (@S1) in the server network (3), distinct from the relay machine (4), [characterized in that it

comprises a first step that associates] the step comprising: associating said address (@S1) with a physical interface (19, 20) of the relay machine (4) that is not linked to the server network (3), so that the relay application (22) receives said datagrams without the need to configure or inform said client application (16) in order to [do so] receive said datagrams.

5. The [Method] method according to claim 4, [characterized in that the first step] wherein the step of associating is preceded by a [second] step [for] of routing the datagrams transmitted through the client network (13), addressed to the server machine (1), to the relay machine (4).

6. The [Relay] relay machine (4) according to claim 1 [or 2], [characterized in that] the application (22) [uses] includes encryption keys [to transmit] and further comprising transmitting encrypted messages received from the network (13) in decrypted fashion inside the network (3).

7. The relay machine (4) according to claim 2, the application (22) includes encryption keys and further comprising transmitting encrypted messages received from the network (13) in decrypted fashion inside the network (3).

[7.] 8. The [Relay] relay machine (4) according to claim 1 [or 2], [characterized in that] the application (22) [uses] includes encryption keys [to transmit] and further comprising transmitting unencrypted messages received from the network (3) in encrypted fashion inside the network (13).

9. The relay machine (4) according to claim 1, the application (22) includes encryption keys and further comprising transmitting unencrypted messages received from the network (3) in encrypted fashion inside the network (13).

ABSTRACT

RELAY FOR ACCESSING A SERVER NETWORK, TRANSPARENT TO A CLIENT NETWORK

[The invention relates to a] A relay machine (4) linked to a client network (13) by means of a first physical interface (19) and linked to a server network (3) by means of a second physical interface (14). The relay machine (4) comprises a first relay application (22) for receiving datagrams addressed to the server machine (1, 2) from the network (13) and for sending to the network (3) datagrams addressed to the server machine (1, 2). An internetwork protocol address (@S1, @S2) of a server machine (1, 2) linked to the server network (3) is associated with the first physical interface (19) so that the datagrams sent up to the application level in the relay machine are available to the relay application in a way that is transparent to the client network (13).

[Fig. 1]

[redacted]
LITERAL TRANSLATION OF Application Based on PCT/FR00/02469 (Serial No.: ________ filed 5/16/01 - Our Ref. T2147-907163

RELAY FOR ACCESSING A SERVER NETWORK, TRANSPARENT TO A CLIENT NETWORK

The technical field to which the invention relates is that of computer networks. Computer networks make it possible to run distributed applications in remote machines linked to the same network or to different networks interconnected by means of interconnection machines.

A transaction between remote machines is initiated by a client application, which sends a request message to a server application in a standby state. The client application places itself in a wait state for a response message to its request message. Upon receiving the request message, the server application generates a response message that it sends to the client application. A network layer allows each message to be conveyed in the form of a datagram, from the machine hosting the sending application to the machine hosting the receiving application. A transport layer allows the message to be conveyed between the sending application and the network layer, then between the network layer and the receiving application, for example from a client application to a server application. An application layer handles the execution of the application in its own environment.

When the machines are not physically linked to the same network, routing protocols of the network layer route the datagrams from the sending machine to an interconnection machine, and from the interconnection machine to the receiving machine, using internetwork protocol addresses, such as for example IP addresses. When passing through the interconnection machine, the datagrams remain at the network layer level. The network between the client machine and the interconnection machine is called the client network. The network between the server machine and the interconnection machine is called the server network.

The technical field to which the invention particularly relates involves an interconnection machine for hosting a relay application, or proxy. A relay application is useful for performing operations on the messages exchanged between the client network and the server network. However, datagrams addressed to the final receiving machine are naturally not sent up to the application layer of the relay machine.

According to the known prior art, the sending application addresses its messages to the relay application of the relay machine instead of addressing them directly to the final receiving application, and indicates in its messages to the relay application the final application to which its messages are to be sent so that the relay application can reroute them by means of the operations it applies to them. This is what happens, for example in an Internet browser, in which it is possible to declare, for a given client application, the address of the relay machine for the network layer and the port number of the relay application for the transport layer, so that the browser encapsulates the address of the server machine and the port number of the final destination application in a datagram addressed to the relay application. However, this makes it necessary to know the relay application through which the messages must pass in order to configure the client machine accordingly. The resulting lack of flexibility, while acceptable for a limited number of applications, is unsatisfactory for a large number of different applications.

The document RFC1928, available on the internet at the address http://www.pmg.lcs.mit.edu/cgi-bin/rfc/view?1928, describes the protocol "SOCKS v5," wherein the port number conventionally used is 1080. Just as for the solution known as "TCP protocol tunneling in web proxy servers," it is necessary to establish a first connection to the relay application, followed by a second connection of the relay machine to the final machine.

In order to eliminate the drawbacks mentioned above, the object of the invention is to allow a client application to simply establish a connection to a server application the way it would when not using the services of a relay application, so that the use of the services of the relay application is transparent for the client application.

A first subject of the invention is a relay machine linked to a client network by means of a first physical interface and linked to a server network by means of a second physical interface, characterized in that at least one internetwork protocol address of a server machine linked to the server network is associated with the first physical interface, and in that it comprises a first relay application for receiving datagrams addressed to the server machine from the client network and for sending to the server network datagrams addressed to the server machine.

Thus, when a datagram arrives in the first physical interface with the internetwork protocol address of the server machine as its destination address, the relay machine is recognized by its network layer as being the destination machine for the datagram. The network layer of the relay machine then sends the datagram up to the application layer of the relay machine by simply following the established protocol. When it receives this datagram, the relay application can process it, after which it may or may not retransmit it to the server machine. This is completely transparent for the client application.

The subject of a variant of the invention is a relay machine linked to a client network by means of a first physical interface and linked to a server network by means of a second physical interface, characterized in that at least one internetwork protocol address of a server machine linked to the server network is associated with a third physical interface, distinct from the first physical interface and from the second physical interface, and in that it comprises a first relay application for receiving datagrams addressed to the server machine from the client network and for sending to the server network datagrams addressed to the server machine.

In this case, the protocol of the network layer does not require the destination address to be assigned to the first physical interface that receives the datagram, but to any physical interface of the relay machine, so that it is sent up to the application layer of the relay machine.

When the relay machine already has a base address in the client network, useful, for example, for routing protocols, said server machine address is associated with the first physical interface as a synonym address of the base address of the relay machine in the client network.

A second subject of the invention is a method for processing, by means of a relay application running in a relay machine between a client network and a server network, datagrams sent through the client network by a client application, addressed to a server machine having an address in the server network, characterized in that it comprises a first step that associates said address in the server network with a physical interface of the relay machine that is not linked to the server network, so that the relay application receives said datagrams.

This offers the advantage of making it unnecessary to configure or inform said client application in order for relay application to be able to process the datagrams. In essence, the client application continues to send its datagrams using the address of the server machine. When the datagram arrives in the first physical interface of the relay machine, the network protocol ensures that the datagram is naturally sent up to the application layer of the relay machine, thus allowing the relay application to receive it.

When it is necessary to route the datagrams transmitted from the client network to the server network through the relay machine, the method is characterized in that the first step is preceded by a second step for routing the datagrams transmitted through the client network, addressed to the server machine, to the relay machine. This is the case, for example, when there is more than one relay machine between the client network and the server network.

Other advantages and details of the implementation of the invention will emerge from the following description in reference to the figures, in which:

- Fig. 1 represents an exemplary relay machine with two physical interfaces;
- Fig. 2 represents an exemplary datagram;
- Fig. 3 represents an exemplary relay machine with three physical interfaces.

In Fig. 1 represents server machines 1, 2 and client machines 11, 12. The machines 1, 2, 11 are linked to a server network 3 by means of respective physical interfaces 7, 8, 17. A client machine 12 is linked to a client network 13 by means of a physical interface 18. The networks 3 and 13 are physically separate. A relay machine 4 is linked to the server network 3 by means of a physical interface 14 and to the network 13 by means of a physical interface 19.

The applications 5, 6, 15, 16 running in the machines 1, 2, 11, 12 communicate with one another through a transport layer CT using a protocol in the connectionless mode such as UDP, or in the connected mode such as TCP. The transport layer CT supervises a network layer CR using a protocol such as IP.

In the network layer CR, the machine 1 is recognized by means of an address @S1, the machine 2 is recognized by means of an address @S2, and the machine 11 is recognized by means of an address @C1. In a known way, each of the addresses @S1, @S2 and @C1 has a network field with a common value that identifies the network 3, and a machine field with a distinct value that identifies each machine linked to the network 3. The machine 12 is recognized by means of an address @C2 with a network field value that identifies the network 13 and a machine field value that identifies the machine 12 in the network 13. The machine 4 is recognized by means of an address @P1 with a network field value that identifies the network 13 and a machine field value that identifies the machine 4 in the network 13, and by means of an address @P2 with a network field value that identifies the network 3 and a machine field value that identifies the machine 4 in the network 3.

The machines communicate with one another by means of messages that flow through the networks in the form of datagrams. Fig. 2 presents an exemplary datagram. This datagram, constituted by a frame of successive bits, is essentially structured in three successive fields. A first field marked DR is dedicated to the protocol of the network layer. A second field marked DT is dedicated to the protocol of the transport layer that supervises the network layer. A third field marked DA is dedicated to an application layer that supervises the transport layer. In the case of a request on the web, for example, the field DR contains the source and destination IP addresses, the field DT contains the source and destination TCP port numbers, and the field DA contains HTTP data.

For example, if a client application 15 running in the client machine 11 issues a request to access a file processed by a server application 5 located in the server machine 1, the application 5 transmits its request to the layer CT of the machine 11, which writes the request into the field DA, and writes into the field DT a service port number for the application 15 and a service port number for the application 5. The layer CT of the machine 11 transmits the fields DT and DA to the layer CR of the machine 11, which writes into the field DR the address @C1 of the machine 11 and the address @S1 of the machine 1. The layer CR then transmits through the interface 17 the datagram thus constituted, which arrives through the interface 7 of the machine 1. The layer CR of the machine 1 recognizes from the address @S1 that the datagram is to be sent to the upper layers of the machine 1, and retransmits the fields DT and DA to the layer CT of the machine 1. Using the service port number for the application 5, the layer CT retransmits the field DA to the application 5, which processes the request.

If an application 16 running in the client machine 12 issues a request to access a file processed by the application 5 located in the server machine 1, the application 16 transmits its request to the layer CT of the machine 12, which writes it into the field DA and which writes into the field DT a service port number for the application 16 and a service port number for the application 5. The layer CT of the machine 12 transmits the fields DT and DA to the layer CR of the machine 12, which writes into the field DR the address @C2 of the machine 12 and the address @S1 of the machine

1. The layer CR then transmits the datagram thus constituted to the interface 18 that arrives through the interface 19 of the machine 4, declared as a router between the networks 13 and 3.

Without the device according to the invention, @S1 not being a destination address of the machine 4, the layer CR of the machine 4 recognizes that the datagram is not to be sent to the upper layers of the machine 4. The layer CR of the machine 4 then searches in routing tables for a line containing a value identical to the network field of the address @S1. The line thus found indicates the interface 14 as being the one for accessing the network 3. The layer CR of the machine 4 therefore retransmits the datagram to the network 3 through the interface 14 so that the datagram arrives through the interface 7 of the machine 1. The layer CR of the machine 1 recognizes from the address @S1 that the datagram is to be sent to the upper layers of the machine 1 and retransmits the fields DT and DA to the layer CT of the machine 1. Using the service port number for the application 5, the layer CT retransmits the field DA to the application 5, which processes the request.

With the device according to the invention, the machine 4 comprises an application 22 that plays the role of a relay, or proxy server, for requests issuing from the network 13. The application 22 offers several advantages; for example, it can control access to the machines 1, 2, 11 linked to the server network 3, it can save responses to previous requests in a cache in order to restore these responses for new requests without requiring these new requests to be routed to the server machine 1, 2.

Several addresses of the layer CR are associated with the physical interface 19, the usual address @P1 and the address @S1 of the server machine 1 linked to the network 3. It is also possible to associate the address @S2 of the server machine 2 with the physical interface 19. As made clear by the description below, unlike the prior art in which it is the client network that determines the utilization of the services of the relay application 22, in this case it is the server network that determines this utilization, for example for accessing the server 1, by associating the address @S1 with the physical interface 19.

The application 22 comprises an input port 9 with the same number as the input port of the application 5, and an output port 10 to which it can assign a number, in order to handle any request messages addressed to the application 5.

As a result of this particular device, the machine 12 does not need to know that it is establishing an intermediate connection with the machine 4. If an application 16 running in the client machine 12 issues a request addressed to the application 5 located in the server machine 1, the address @S1 is then recognized in the network 13 as being the address of the machine 4.

In order to issue a request addressed to the application 5, the application 16 sends a datagram Q through the network 13 that contains the addresses @S1 and @C2 in the field CR, the port numbers of the applications 5 an 16 in the transport field, and the final information addressed to the application 5 in the field CA.

When the datagram Q is received through the physical interface 19 of the machine 4, the network layer CR of the machine 4 recognizes the destination address @S1 in the field DR as being an address that belongs to it, and therefore sends the datagram up to the transport layer CT of the machine 4. The transport layer CT recognizes the destination number in the field DT as being the number of the port 9 of the application 22, to which it then transmits the content of the datagram Q.

The application 22 then processes the content of the field DA of the datagram Q. The processing of the datagram Q by the application 22 consists, for example, of verifying access rights, and checking to see if the machine 4 already contains a response to the request in its cache in order to decide whether or not to communicate the datagram Q to the server application 5.

When, in order to process the request message received from the client application 16, the application 22 needs to send a request message to the application 5, the application 22 communicates the following data to the transport layer CT of the machine 4: the content of the request to be entered into the field DA, the input port number of the application 5, an output port number of the application 22 for handling the response to the request, and the internetwork protocol address @S1 of the machine 1. These data are transmitted to the network layer CR of the machine 4. Upon receiving these data, the network layer CR of the machine 4 searches in its routing tables for the network through which to send a datagram, based on the network field of the address @S1. In the example described here, the network field of the address @S1 corresponding to the network 3 to which the machine 1 is linked, the layer CR sends to the physical interface 14 a datagram containing in the field DR the destination address @S1and the source address @P2 associated with the physical interface 14. In the server network 3, the datagram conventionally reaches the machine 1 and the server application 5 in the machine 1.

The response received from the application 5 through the interface 14 is sent to the application 22 by the network layer because the address @P2 is an address of the machine 4, and by the transport layer CT because the port number for the response is the one assigned to the port 10 by the application 22. Using an internal request and response handling mechanism, the application 22 associates the response with the outgoing port number received from the application 16. In order to retransmit the response to the application 16, the application 22 communicates the following data to the transport layer CT of the machine 4: the content of the response to be entered into the field DA, the output port number of the application 16, the input port number of the application 22 which is identical to the input port number of the application 5 for handling the response to the request, the destination internetwork protocol address @C2 of the machine 12 and the source internetwork protocol address @S1 of the machine 1. These data are transmitted to the network layer CR of the machine 4 by the transport layer. Upon receiving these data, the network layer CR of the machine 4 searches in its routing tables for the network to which to send a datagram, based on the network field of the address @C2. In the example described here, the network field of the address @C2 corresponding to the network 13 to which the machine 12 is linked, the layer CR sends to the physical interface 19 a datagram that contains, in the field DR, the destination address @P2 and the source address @S1 associated with the physical interface 19. In the client network 13, the datagram conventionally reaches the machine 12 and the client application 16 in the machine 12.

Thus, the application 16 in the machine 12 receives a response that is returned by the application 5 in the machine 1 without having to pass through the application 22; this occurs in a way that is transparent for the client application 16.

Referring to Fig. 3, the address @S1 is associated with a physical interface 20 that is different both from the interface 14 as in the preceding case, and from the interface 19 as in this particular case.

When a datagram is sent through the network 13 with the address @S1, the routing protocol of the network layer CR of the machine 4 detects it in the interface 19 with which the address @P1 is associated. Since the address @S1 associated with the physical interface 20 is an address of the machine 4, the datagram is sent up to the application layer CA of the machine 4.

A relay application 21 processes the request message obtained from the datagram received, just like the preceding relay application 22. In order to send the response message to the application 12, the relay application 22 has a specific driver to a virtual network to which the physical interface 20 is linked.

The case in which the IP address @S1 is associated with the interface 19 is particularly advantageous for making the invention easy to use. In the simple example that follows, the application 16 executes a Telnet function as a client application, and the application 22 executes a telnetd function as a server application of the application 16 and a Telnet function as a client of the application 5. The application 5 executes a telnetd function as a server of the application 22. Telnet and telnetd are known functions that use TCP/IP to connect a terminal of a client machine in which the Telnet function is executed to a server machine in which the telnetd function is executed.

In order to keep track of the machine in which the commands are executed, each machine runs on a different operating system. The client machine 12 runs on an AIX (registered trademark) version 4.1 system, and has the IP address @C1 = 129.182.51.58. The relay machine 4 runs on an AIX version 4.2 system and has the IP addresses @P1 = 129.182.51.21 and @P2 = 192.90.249.22. The server machine 12 runs on a (proprietary) DNS-E system and has the IP address @S1 = 192.90.249.124. The network 13 is accessible in a known way at an IP address @R1 = 129.182.50 with a mask @M1 = 255.255.254.0.

In the client machine 12, the command

```
route add –host 192.90.249.124 129.182.51.21
```

means that in order to reach the server machine 1 with the address @S1, the datagrams sent pass through the relay machine with the address @P1.

In the server machine 1, the command

```
route add –net 129.182.50 192.90.249.22 –netmask 255.255.254.0
```

means that in order to reach any machine of the network 13 with the address @R1, the datagrams sent pass through the relay machine with the address @P2.

In the client machine 12, the command

```
Telnet 192.90.249.124
```

activates the Telnet application in order to reach the server machine 1 with the address @S1. At this stage, the only machine recognized through the IP address @S1 is the server machine 1. The IP layer of the machine 4 routes the datagrams sent by the IP layer of the machine 12 to the IP layer of the server machine 1. The IP layer of the machine 1, recognizing the address @S1, sends the application field of the datagrams to the telnetd application of the machine 1. In return, the telnetd application of the machine 1 sends the machine 12 the message:

```
Trying...
Connected to 192.90.249.124.
Escape character is '^]'.
$$ 0000 *DNS-E V3U1.000 P1.001 P2.019 P3.010*IMA:BX77SIM
1998/10/21 17:23*
```

The display of this message on the terminal of the machine 12 shows that it is in the DNS system environment, which means that the machine 1 has been reached directly. The relay machine 4 was not passed through in order to perform the IP routing.

In the client machine 12, the command

```
Telnet 129.182.51.21
```

activates the Telnet application in order to reach the relay machine 4 with the address @P1. The IP layer of the machine 4, recognizing the address @P1, sends the application field of the datagrams to the telnetd application of the machine 4. In return, the telnetd application of the machine 4 sends the machine 12 the message

```
Trying...
Connected to 129.182.51.21.
Escape character is '^]'.
Telnet (thirteen)
AIX Version 4
© Copyrights by IBM and by others 1982, 1996.
Login:
```

The display of this message on the terminal of the machine 12 shows that it is in the AIX system environment, which means that the machine 4 has been reached.

This makes it possible to generate commands from the terminal of the machine 12 that are executed in the machinè 4.

In the machine 4, the interface 19 being named en1, the command:

```
ifconfig en1 192.90.249.124 alias
```

defines the address @S1 as an additional address associated with the interface 19. The machine 4 runs no risk of being confused with the machine 1 in the network 13 by the IP layer, since it is physically separate from the network 3. Likewise, the command:

```
ifconfig en1 192.90.249.125 alias
```

would define the address @S2 as an additional address associated with the interface 19.

Referring again to the machine 12, the command:

```
Telnet 192.90.249.124
```

activates the Telnet application with an effect that is different than the one described above. The message displayed on the terminal of the machine 12 is:

```
Trying...
Connected to 129.182.51.21.
Escape character is '^]'.
Telnet (thirteen)
AIX Version 4
© Copyrights by IBM and by others 1982, 1996.
Login:
```

The display of this message on the terminal of the machine 12 shows that the latter is in the AIX system environment of the machine 4. Despite having requested a connection to the telnetd application of the server machine 1 using the address @S1, the command has established a connection with the telnetd application of the machine 4. This is explained by the fact that the IP layer of the machine 4 recognizes the address @S1 as a destination address belonging to the machine 4, without taking into account the routing through the network 3. Thus, the IP layer of the machine 4 sends the application field of the datagrams received through the interface 19 to the telnetd application of the machine 4.

At present, in the machine 4, the command:

```
Telnet 192.90.249.124
```

activates the Telnet application in order to reach the server machine 1 with the address @S1. At this stage, the only machine recognized by the IP address @S1 from the interface 14 is the server machine 1. The IP layer of the machine 1, recognizing the address @S1, sends the application field of the datagrams up to the telnetd application of the machine 1. In return, the telnetd application of the machine 1 sends to the Telnet application of the machine 4 the message:

```
Trying...
Connected to 192.90.249.124.
Escape character is '^]'.
$$ 0000 *DNS-E V3U1.000 P1.001 P2.019 P3.010*IMA:BX77SIM
1998/10/21 17:23*
```

This message is retransmitted by the telnetd application of the machine 4 to the Telnet application of the machine 12. The display of this message on the terminal of the machine 12 shows that it is in the DNS system environment, i.e., that the machine 1 has been reached. However, the application field of the datagrams is sent up to the application layer of the relay machine 4 in a way that is transparent for the machine 12.

The method explained above in terms of a manual operation can be implemented by means of a program executed by the application layer of the machine 4.

The datagrams addressed to the machine 1, which pass through the IP layer of the machine 4, are sent up to the application layer of the machine 4 because the address @S1 is associated with a physical interface of the machine 4. In order to avoid conflicts in the network 3 with the machine 1, it is preferable not to associate the address @S1 with the interface 14. Referring to Fig. 3, it is possible to associate the address @S1 with a physical interface other than the interface 19, for example a physical interface 20.

One example of a particular operation by the application 22 described here offers a particular advantage. If encryption keys are associated with the address @S1 in order to encrypt the requests received from and the responses sent to the machine 12, the decryption of the requests and the encryption of the responses can be handled by the machine 4. The decrypted data can flow through the server network 3 without any risk. Thus, the encryption and decryption resources can be centralized in the machine 4, leaving a maximum number of resources available in the machine 1 for its server functions. The application 22 is also responsible for re-encrypting the responses prior to sending them through the network 13.

CLAIMS

1. Relay machine (4) linked to a client network (13) by means of a first physical interface (19) and linked to a server network (3) by means of a second physical interface (14), characterized in that at least one internetwork protocol address (@S1, @S2) of a server machine (1, 2) linked to the server network (3), distinct from the relay machine (4), is associated with the first physical interface (19), and in that it comprises a first relay application (22) for receiving datagrams addressed to the server machine (1, 2) from the client network (13) and for sending to the server network (3) datagrams addressed to the server machine (1, 2).

2. Relay machine (4) linked to a client network (13) by means of a first physical interface (19) and linked to a server network (3) by means of a second physical interface (14), characterized in that at least one internetwork protocol address (@S1, @S2) of a server machine (1, 2) linked to the server network (3), distinct from the relay machine (4), is associated with a third physical interface (20), distinct from the first physical interface (19) and from the second physical interface (14), and in that it comprises a first relay application (22) for receiving datagrams addressed to the server machine (1, 2) from the client network (13) and for sending to the server network (3) datagrams addressed to the server machine (1, 2).

3. Relay machine (4) according to claim 1, characterized in that said address (@S1, @S2) is associated with the first physical interface (19) as an address synonymous with a base address (@P1) of the machine (4) in the network (13).

4. Method for processing, by means of at least one relay application (22) running in a relay machine (4) between a client network (13) and a server network (3), datagrams sent through the client network (13) by a client application (16) to a server machine (1) with the address (@S1) in the server network (3), distinct from the relay machine (4), characterized in that it comprises a first step that associates said address (@S1) with a physical interface (19, 20) of the relay machine (4) that is not linked to the server network (3), so that the relay application (22) receives said datagrams without the need to configure or inform said client application (16) in order to do so.

5. Method according to claim 4, characterized in that the first step is preceded by a second step for routing the datagrams transmitted through the client network (13), addressed to the server machine (1), to the relay machine (4).

6. Relay machine (4) according to claim 1 or 2, characterized in that the application (22) uses encryption keys to transmit encrypted messages received from the network (13) in decrypted fashion inside the network (3).

7. Relay machine (4) according to claim 1 or 2, characterized in that the application (22) uses encryption keys to transmit unencrypted messages received from the network (3) in encrypted fashion inside the network (13).

ABSTRACT

RELAY FOR ACCESSING A SERVER NETWORK, TRANSPARENT TO A CLIENT NETWORK

The invention relates to a relay machine (4) linked to a client network (13) by means of a first physical interface (19) and linked to a server network (3) by means of a second physical interface (14). The relay machine (4) comprises a first relay application (22) for receiving datagrams addressed to the server machine (1, 2) from the network (13) and for sending to the network (3) datagrams addressed to the server machine (1, 2). An internetwork protocol address (@S1, @S2) of a server machine (1, 2) linked to the server network (3) is associated with the first physical interface (19) so that the datagrams sent up to the application level in the relay machine are available to the relay application in a way that is transparent to the client network (13).

Fig. 1

Fig.1

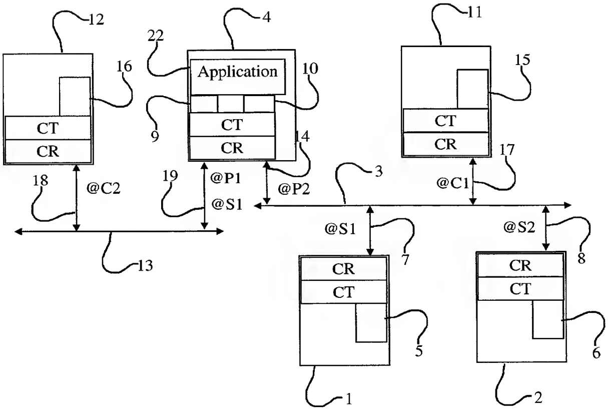

12
22
4
11
16
Application
10
15
CT
CR
9
CT
CR
14
CT
CR
17
18
@C2
19
@P1
@P2
3
@C1
@S1
@S1
@S2
13
CR
CT
7
CR
CT
8
5
6
1
2

Fig.2

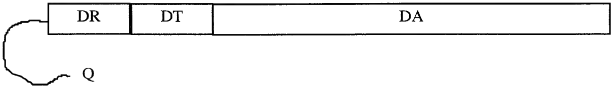
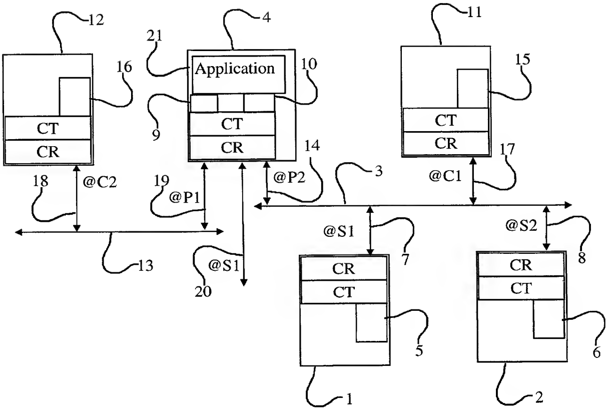

DR
DT
DA
Q

Fig.3


12
4
11
21
16
Application
10
15
CT
CR
CT
CR
CT
CR
9
14
17
18
@C2
19
@P1
@P2
3
@C1
@S1
@S2
7
8
13
@S1
20
CR
CT
CR
CT
5
6
1
2

RELAY FOR ACCESSING A SERVER NETWORK, TRANSPARENT TO A CLIENT NETWORK

The technical field to which the invention relates is that of computer networks. Computer networks make it possible to run distributed applications in remote machines linked to the same network or to different networks interconnected by means of interconnection machines.

A transaction between remote machines is initiated by a client application, which sends a request message to a server application in a standby state. The client application places itself in a wait state for a response message to its request message. Upon receiving the request message, the server application generates a response message that it sends to the client application. A network layer allows each message to be conveyed in the form of a datagram, from the machine hosting the sending application to the machine hosting the receiving application. A transport layer allows the message to be conveyed between the sending application and the network layer, then between the network layer and the receiving application, for example from a client application to a server application. An application layer handles the execution of the application in its own environment.

When the machines are not physically linked to the same network, routing protocols of the network layer route the datagrams from the sending machine to an interconnection machine, and from the interconnection machine to the receiving machine, using internetwork protocol addresses, such as for example IP addresses. When passing through the interconnection machine, the datagrams remain at the network layer level. The network between the client machine and the interconnection machine is called the client network. The network between the server machine and the interconnection machine is called the server network.

The technical field to which the invention particularly relates involves an interconnection machine for hosting a relay application, or proxy. A relay application is useful for performing operations on the messages exchanged between the client network and the server network. However, datagrams addressed to the final receiving machine are naturally not sent up to the application layer of the relay machine.

According to the known prior art, the sending application addresses its messages to the relay application of the relay machine instead of addressing them directly to the final receiving application, and indicates in its messages to the relay application the final application to which its messages are to be sent so that the relay application can reroute them by means of the operations it applies to them. This is what happens, for example in an Internet browser, in which it is possible to declare, for a given client application, the address of the relay machine for the network layer and the port number of the relay application for the transport layer, so that the browser encapsulates the address of the server machine and the port number of the final destination application in a datagram addressed to the relay application. However, this makes it necessary to know the relay application through which the messages must pass in order to configure the client machine accordingly. The resulting lack of flexibility, while acceptable for a limited number of applications, is unsatisfactory for a large number of different applications.

The document RFC1928, available on the internet at the address http://www.pmg.lcs.mit.edu/cgi-bin/rfc/view?1928, describes the protocol "SOCKS v5," wherein the port number conventionally used is 1080. Just as for the solution known as "TCP protocol tunneling in web proxy servers," it is necessary to establish a first connection to the relay application, followed by a second connection of the relay machine to the final machine.

In order to eliminate the drawbacks mentioned above, the object of the invention is to allow a client application to simply establish a connection to a server application the way it would when not using the services of a relay application, so that the use of the services of the relay application is transparent for the client application.

A first subject of the invention is a relay machine linked to a client network by means of a first physical interface and linked to a server network by means of a second physical interface, characterized in that at least one internetwork protocol address of a server machine linked to the server network is associated with the first physical interface, and in that it comprises a first relay application for receiving datagrams addressed to the server machine from the client network and for sending to the server network datagrams addressed to the server machine.

Thus, when a datagram arrives in the first physical interface with the internetwork protocol address of the server machine as its destination address, the relay machine is recognized by its network layer as being the destination machine for the datagram. The network layer of the relay machine then sends the datagram up to the application layer of the relay machine by simply following the established protocol. When it receives this datagram, the relay application can process it, after which it may or may not retransmit it to the server machine. This is completely transparent for the client application.

The subject of a variant of the invention is a relay machine linked to a client network by means of a first physical interface and linked to a server network by means of a second physical interface, characterized in that at least one internetwork protocol address of a server machine linked to the server network is associated with a third physical interface, distinct from the first physical interface and from the second physical interface, and in that it comprises a first relay application for receiving datagrams addressed to the server machine from the client network and for sending to the server network datagrams addressed to the server machine.

In this case, the protocol of the network layer does not require the destination address to be assigned to the first physical interface that receives the datagram, but to any physical interface of the relay machine, so that it is sent up to the application layer of the relay machine.

When the relay machine already has a base address in the client network, useful, for example, for routing protocols, said server machine address is associated with the first physical interface as a synonym address of the base address of the relay machine in the client network.

A second subject of the invention is a method for processing, by means of a relay application running in a relay machine between a client network and a server network, datagrams sent through the client network by a client application, addressed to a server machine having an address in the server network, characterized in that it comprises a first step that associates said address in the server network with a physical interface of the relay machine that is not linked to the server network, so that the relay application receives said datagrams.

This offers the advantage of making it unnecessary to configure or inform said client application in order for relay application to be able to process the datagrams. In essence, the client application continues to send its datagrams using the address of the server machine. When the datagram arrives in the first physical interface of the relay machine, the network protocol ensures that the datagram is naturally sent up to the application layer of the relay machine, thus allowing the relay application to receive it.

When it is necessary to route the datagrams transmitted from the client network to the server network through the relay machine, the method is characterized in that the first step is preceded by a second step for routing the datagrams transmitted through the client network, addressed to the server machine, to the relay machine. This is the case, for example, when there is more than one relay machine between the client network and the server network.

Other advantages and details of the implementation of the invention will emerge from the following description in reference to the figures, in which:

- Fig. 1 represents an exemplary relay machine with two physical interfaces;
- Fig. 2 represents an exemplary datagram;
- Fig. 3 represents an exemplary relay machine with three physical interfaces.

In Fig. 1 represents server machines 1, 2 and client machines 11, 12. The machines 1, 2, 11 are linked to a server network 3 by means of respective physical interfaces 7, 8, 17. A client machine 12 is linked to a client network 13 by means of a physical interface 18. The networks 3 and 13 are physically separate. A relay machine 4 is linked to the server network 3 by means of a physical interface 14 and to the network 13 by means of a physical interface 19.

The applications 5, 6, 15, 16 running in the machines 1, 2, 11, 12 communicate with one another through a transport layer CT using a protocol in the connectionless mode such as UDP, or in the connected mode such as TCP. The transport layer CT supervises a network layer CR using a protocol such as IP.

In the network layer CR, the machine 1 is recognized by means of an address @S1, the machine 2 is recognized by means of an address @S2, and the machine 11 is recognized by means of an address @C1. In a known way, each of the addresses @S1, @S2 and @C1 has a network field with a common value that identifies the network 3, and a machine field with a distinct value that identifies each machine linked to the network 3. The machine 12 is recognized by means of an address @C2 with a network field value that identifies the network 13 and a machine field value that identifies the machine 12 in the network 13. The machine 4 is recognized by means of an address @P1 with a network field value that identifies the network 13 and a machine field value that identifies the machine 4 in the network 13, and by means of an address @P2 with a network field value that identifies the network 3 and a machine field value that identifies the machine 4 in the network 3.

The machines communicate with one another by means of messages that flow through the networks in the form of datagrams. Fig. 2 presents an exemplary datagram. This datagram, constituted by a frame of successive bits, is essentially structured in three successive fields. A first field marked DR is dedicated to the protocol of the network layer. A second field marked DT is dedicated to the protocol of the transport layer that supervises the network layer. A third field marked DA is dedicated to an application layer that supervises the transport layer. In the case of a request on the web, for example, the field DR contains the source and destination IP addresses, the field DT contains the source and destination TCP port numbers, and the field DA contains HTTP data.

For example, if a client application 15 running in the client machine 11 issues a request to access a file processed by a server application 5 located in the server machine 1, the application 5 transmits its request to the layer CT of the machine 11, which writes the request into the field DA, and writes into the field DT a service port number for the application 15 and a service port number for the application 5. The layer CT of the machine 11 transmits the fields DT and DA to the layer CR of the machine 11, which writes into the field DR the address @C1 of the machine 11 and the address @S1 of the machine 1. The layer CR then transmits through the interface 17 the datagram thus constituted, which arrives through the interface 7 of the machine 1. The layer CR of the machine 1 recognizes from the address @S1 that the datagram is to be sent to the upper layers of the machine 1, and retransmits the fields DT and DA to the layer CT of the machine 1. Using the service port number for the application 5, the layer CT retransmits the field DA to the application 5, which processes the request.

If an application 16 running in the client machine 12 issues a request to access a file processed by the application 5 located in the server machine 1, the application 16 transmits its request to the layer CT of the machine 12, which writes it into the field DA and which writes into the field DT a service port number for the application 16 and a service port number for the application 5. The layer CT of the machine 12 transmits the fields DT and DA to the layer CR of the machine 12, which writes into the field DR the address @C2 of the machine 12 and the address @S1 of the machine

1. The layer CR then transmits the datagram thus constituted to the interface 18 that arrives through the interface 19 of the machine 4, declared as a router between the networks 13 and 3.

Without the device according to the invention, @S1 not being a destination address of the machine 4, the layer CR of the machine 4 recognizes that the datagram is not to be sent to the upper layers of the machine 4. The layer CR of the machine 4 then searches in routing tables for a line containing a value identical to the network field of the address @S1. The line thus found indicates the interface 14 as being the one for accessing the network 3. The layer CR of the machine 4 therefore retransmits the datagram to the network 3 through the interface 14 so that the datagram arrives through the interface 7 of the machine 1. The layer CR of the machine 1 recognizes from the address @S1 that the datagram is to be sent to the upper layers of the machine 1 and retransmits the fields DT and DA to the layer CT of the machine 1. Using the service port number for the application 5, the layer CT retransmits the field DA to the application 5, which processes the request.

With the device according to the invention, the machine 4 comprises an application 22 that plays the role of a relay, or proxy server, for requests issuing from the network 13. The application 22 offers several advantages; for example, it can control access to the machines 1, 2, 11 linked to the server network 3, it can save responses to previous requests in a cache in order to restore these responses for new requests without requiring these new requests to be routed to the server machine 1, 2.

Several addresses of the layer CR are associated with the physical interface 19, the usual address @P1 and the address @S1 of the server machine 1 linked to the network 3. It is also possible to associate the address @S2 of the server machine 2 with the physical interface 19. As made clear by the description below, unlike the prior art in which it is the client network that determines the utilization of the services of the relay application 22, in this case it is the server network that determines this utilization, for example for accessing the server 1, by associating the address @S1 with the physical interface 19.

The application 22 comprises an input port 9 with the same number as the input port of the application 5, and an output port 10 to which it can assign a number, in order to handle any request messages addressed to the application 5.

As a result of this particular device, the machine 12 does not need to know that it is establishing an intermediate connection with the machine 4. If an application 16 running in the client machine 12 issues a request addressed to the application 5 located in the server machine 1, the address @S1 is then recognized in the network 13 as being the address of the machine 4.

In order to issue a request addressed to the application 5, the application 16 sends a datagram Q through the network 13 that contains the addresses @S1 and @C2 in the field CR, the port numbers of the applications 5 an 16 in the transport field, and the final information addressed to the application 5 in the field CA.

When the datagram Q is received through the physical interface 19 of the machine 4, the network layer CR of the machine 4 recognizes the destination address @S1 in the field DR as being an address that belongs to it, and therefore sends the datagram up to the transport layer CT of the machine 4. The transport layer CT recognizes the destination number in the field DT as being the number of the port 9 of the application 22, to which it then transmits the content of the datagram Q.

The application 22 then processes the content of the field DA of the datagram Q. The processing of the datagram Q by the application 22 consists, for example, of verifying access rights, and checking to see if the machine 4 already contains a response to the request in its cache in order to decide whether or not to communicate the datagram Q to the server application 5.

When, in order to process the request message received from the client application 16, the application 22 needs to send a request message to the application 5, the application 22 communicates the following data to the transport layer CT of the machine 4: the content of the request to be entered into the field DA, the input port number of the application 5, an output port number of the application 22 for handling the response to the request, and the internetwork protocol address @S1 of the machine 1. These data are transmitted to the network layer CR of the machine 4. Upon receiving these data, the network layer CR of the machine 4 searches in its routing tables for the network through which to send a datagram, based on the network field of the address @S1. In the example described here, the network field of the address @S1 corresponding to the network 3 to which the machine 1 is linked, the layer CR sends to the physical interface 14 a datagram containing in the field DR the destination address @S1and the source address @P2 associated with the physical interface 14. In the server network 3, the datagram conventionally reaches the machine 1 and the server application 5 in the machine 1.

The response received from the application 5 through the interface 14 is sent to the application 22 by the network layer because the address @P2 is an address of the machine 4, and by the transport layer CT because the port number for the response is the one assigned to the port 10 by the application 22. Using an internal request and response handling mechanism, the application 22 associates the response with the outgoing port number received from the application 16. In order to retransmit the response to the application 16, the application 22 communicates the following data to the transport layer CT of the machine 4: the content of the response to be entered into the field DA, the output port number of the application 16, the input port number of the application 22 which is identical to the input port number of the application 5 for handling the response to the request, the destination internetwork protocol address @C2 of the machine 12 and the source internetwork protocol address @S1 of the machine 1. These data are transmitted to the network layer CR of the machine 4 by the transport layer. Upon receiving these data, the network layer CR of the machine 4 searches in its routing tables for the network to which to send a datagram, based on the network field of the address @C2. In the example described here, the network field of the address @C2 corresponding to the network 13 to which the machine 12 is linked, the layer CR sends to the physical interface 19 a datagram that contains, in the field DR, the destination address @P2 and the source address @S1 associated with the physical interface 19. In the client network 13, the datagram conventionally reaches the machine 12 and the client application 16 in the machine 12.

Thus, the application 16 in the machine 12 receives a response that is returned by the application 5 in the machine 1 without having to pass through the application 22; this occurs in a way that is transparent for the client application 16.

Referring to Fig. 3, the address @S1 is associated with a physical interface 20 that is different both from the interface 14 as in the preceding case, and from the interface 19 as in this particular case.

When a datagram is sent through the network 13 with the address @S1, the routing protocol of the network layer CR of the machine 4 detects it in the interface 19 with which the address @P1 is associated. Since the address @S1 associated with the physical interface 20 is an address of the machine 4, the datagram is sent up to the application layer CA of the machine 4.

A relay application 21 processes the request message obtained from the datagram received, just like the preceding relay application 22. In order to send the response message to the application 12, the relay application 22 has a specific driver to a virtual network to which the physical interface 20 is linked.

The case in which the IP address @S1 is associated with the interface 19 is particularly advantageous for making the invention easy to use. In the simple example that follows, the application 16 executes a Telnet function as a client application, and the application 22 executes a telnetd function as a server application of the application 16 and a Telnet function as a client of the application 5. The application 5 executes a telnetd function as a server of the application 22. Telnet and telnetd are known functions that use TCP/IP to connect a terminal of a client machine in which the Telnet function is executed to a server machine in which the telnetd function is executed.

In order to keep track of the machine in which the commands are executed, each machine runs on a different operating system. The client machine 12 runs on an AIX (registered trademark) version 4.1 system, and has the IP address @C1 = 129.182.51.58. The relay machine 4 runs on an AIX version 4.2 system and has the IP addresses @P1 = 129.182.51.21 and @P2 = 192.90.249.22. The server machine 12 runs on a (proprietary) DNS-E system and has the IP address @S1 = 192.90.249.124. The network 13 is accessible in a known way at an IP address @R1 = 129.182.50 with a mask @M1 = 255.255.254.0.

In the client machine 12, the command

```
route add –host 192.90.249.124 129.182.51.21
```

means that in order to reach the server machine 1 with the address @S1, the datagrams sent pass through the relay machine with the address @P1.

In the server machine 1, the command

```
route add –net 129.182.50 192.90.249.22 –netmask 255.255.254.0
```

means that in order to reach any machine of the network 13 with the address @R1, the datagrams sent pass through the relay machine with the address @P2.

In the client machine 12, the command

```
Telnet 192.90.249.124
```

activates the Telnet application in order to reach the server machine 1 with the address @S1. At this stage, the only machine recognized through the IP address @S1 is the server machine 1. The IP layer of the machine 4 routes the datagrams sent by the IP layer of the machine 12 to the IP layer of the server machine 1. The IP layer of the machine 1, recognizing the address @S1, sends the application field of the datagrams to the telnetd application of the machine 1. In return, the telnetd application of the machine 1 sends the machine 12 the message:

```
Trying...
Connected to 192.90.249.124.
Escape character is '^]'.
$$ 0000 *DNS-E V3U1.000 P1.001 P2.019 P3.010*IMA:BX77SIM
1998/10/21 17:23*
```

The display of this message on the terminal of the machine 12 shows that it is in the DNS system environment, which means that the machine 1 has been reached directly. The relay machine 4 was not passed through in order to perform the IP routing.

In the client machine 12, the command

```
Telnet 129.182.51.21
```

activates the Telnet application in order to reach the relay machine 4 with the address @P1. The IP layer of the machine 4, recognizing the address @P1, sends the application field of the datagrams to the telnetd application of the machine 4. In return, the telnetd application of the machine 4 sends the machine 12 the message

```
Trying...
Connected to 129.182.51.21.
Escape character is '^]'.
Telnet (thirteen)
AIX Version 4
© Copyrights by IBM and by others 1982, 1996.
Login:
```

The display of this message on the terminal of the machine 12 shows that it is in the AIX system environment, which means that the machine 4 has been reached.

This makes it possible to generate commands from the terminal of the machine 12 that are executed in the machine 4.

In the machine 4, the interface 19 being named en1, the command:

```
ifconfig en1 192.90.249.124 alias
```

defines the address @S1 as an additional address associated with the interface 19. The machine 4 runs no risk of being confused with the machine 1 in the network 13 by the IP layer, since it is physically separate from the network 3. Likewise, the command:

```
ifconfig en1 192.90.249.125 alias
```

would define the address @S2 as an additional address associated with the interface 19.

Referring again to the machine 12, the command:

```
Telnet 192.90.249.124
```

activates the Telnet application with an effect that is different than the one described above. The message displayed on the terminal of the machine 12 is:

```
Trying...
Connected to 129.182.51.21.
Escape character is '^]'.
Telnet (thirteen)
AIX Version 4
© Copyrights by IBM and by others 1982, 1996.
Login:
```

The display of this message on the terminal of the machine 12 shows that the latter is in the AIX system environment of the machine 4. Despite having requested a connection to the telnetd application of the server machine 1 using the address @S1, the command has established a connection with the telnetd application of the machine 4. This is explained by the fact that the IP layer of the machine 4 recognizes the address @S1 as a destination address belonging to the machine 4, without taking into account the routing through the network 3. Thus, the IP layer of the machine 4 sends the application field of the datagrams received through the interface 19 to the telnetd application of the machine 4.

At present, in the machine 4, the command:

```
Telnet 192.90.249.124
```

activates the Telnet application in order to reach the server machine 1 with the address @S1. At this stage, the only machine recognized by the IP address @S1 from the interface 14 is the server machine 1. The IP layer of the machine 1, recognizing the address @S1, sends the application field of the datagrams up to the telnetd application of the machine 1. In return, the telnetd application of the machine 1 sends to the Telnet application of the machine 4 the message:

```
Trying...
Connected to 192.90.249.124.
Escape character is '^]'.
$$ 0000 *DNS-E V3U1.000 P1.001 P2.019 P3.010*IMA:BX77SIM
1998/10/21 17:23*
```

This message is retransmitted by the telnetd application of the machine 4 to the Telnet application of the machine 12. The display of this message on the terminal of the machine 12 shows that it is in the DNS system environment, i.e., that the machine 1 has been reached. However, the application field of the datagrams is sent up to the application layer of the relay machine 4 in a way that is transparent for the machine 12.

The method explained above in terms of a manual operation can be implemented by means of a program executed by the application layer of the machine 4.

The datagrams addressed to the machine 1, which pass through the IP layer of the machine 4, are sent up to the application layer of the machine 4 because the address @S1 is associated with a physical interface of the machine 4. In order to avoid conflicts in the network 3 with the machine 1, it is preferable not to associate the address @S1 with the interface 14. Referring to Fig. 3, it is possible to associate the address @S1 with a physical interface other than the interface 19, for example a physical interface 20.

One example of a particular operation by the application 22 described here offers a particular advantage. If encryption keys are associated with the address @S1 in order to encrypt the requests received from and the responses sent to the machine 12, the decryption of the requests and the encryption of the responses can be handled by the machine 4. The decrypted data can flow through the server network 3 without any risk. Thus, the encryption and decryption resources can be centralized in the machine 4, leaving a maximum number of resources available in the machine 1 for its server functions. The application 22 is also responsible for re-encrypting the responses prior to sending them through the network 13.

CLAIMS

1. Relay machine (4) linked to a client network (13) by means of a first physical interface (19) and linked to a server network (3) by means of a second physical interface (14), characterized in that at least one internetwork protocol address (@S1, @S2) of a server machine (1, 2) linked to the server network (3), distinct from the relay machine (4), is associated with the first physical interface (19), and in that it comprises a first relay application (22) for receiving datagrams addressed to the server machine (1, 2) from the client network (13) and for sending to the server network (3) datagrams addressed to the server machine (1, 2).

2. Relay machine (4) linked to a client network (13) by means of a first physical interface (19) and linked to a server network (3) by means of a second physical interface (14), characterized in that at least one internetwork protocol address (@S1, @S2) of a server machine (1, 2) linked to the server network (3), distinct from the relay machine (4), is associated with a third physical interface (20), distinct from the first physical interface (19) and from the second physical interface (14), and in that it comprises a first relay application (22) for receiving datagrams addressed to the server machine (1, 2) from the client network (13) and for sending to the server network (3) datagrams addressed to the server machine (1, 2).

3. Relay machine (4) according to claim 1, characterized in that said address (@S1, @S2) is associated with the first physical interface (19) as an address synonymous with a base address (@P1) of the machine (4) in the network (13).

4. Method for processing, by means of at least one relay application (22) running in a relay machine (4) between a client network (13) and a server network (3), datagrams sent through the client network (13) by a client application (16) to a server machine (1) with the address (@S1) in the server network (3), distinct from the relay machine (4), characterized in that it comprises a first step that associates said address (@S1) with a physical interface (19, 20) of the relay machine (4) that is not linked to the server network (3), so that the relay application (22) receives said datagrams without the need to configure or inform said client application (16) in order to do so.

5. Method according to claim 4, characterized in that the first step is preceded by a second step for routing the datagrams transmitted through the client network (13), addressed to the server machine (1), to the relay machine (4).

6. Relay machine (4) according to claim 1 or 2, characterized in that the application (22) uses encryption keys to transmit encrypted messages received from the network (13) in decrypted fashion inside the network (3).

7. Relay machine (4) according to claim 1 or 2, characterized in that the application (22) uses encryption keys to transmit unencrypted messages received from the network (3) in encrypted fashion inside the network (13).

ABSTRACT

RELAY FOR ACCESSING A SERVER NETWORK, TRANSPARENT TO A CLIENT NETWORK

The invention relates to a relay machine (4) linked to a client network (13) by means of a first physical interface (19) and linked to a server network (3) by means of a second physical interface (14). The relay machine (4) comprises a first relay application (22) for receiving datagrams addressed to the server machine (1, 2) from the network (13) and for sending to the network (3) datagrams addressed to the server machine (1, 2). An internetwork protocol address (@S1, @S2) of a server machine (1, 2) linked to the server network (3) is associated with the first physical interface (19) so that the datagrams sent up to the application level in the relay machine are available to the relay application in a way that is transparent to the client network (13).

Fig. 1

Fig.1

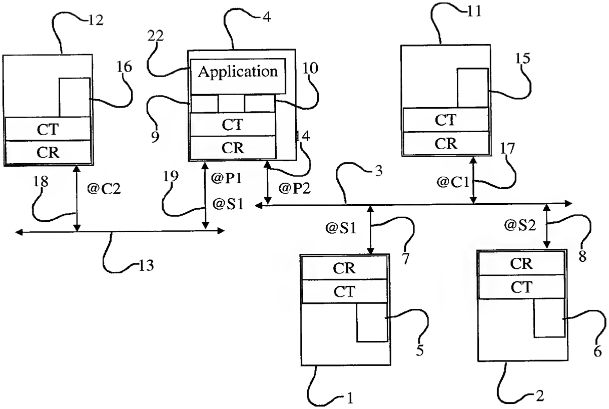

12
22
4
11
16
Application
10
15
CT
CR
9
CT
CR
14
CT
CR
17
18
@C2
19
@P1
@P2
3
@C1
@S1
@S1
@S2
7
8
13
CR
CT
CR
CT
5
6
1
2

Fig.2

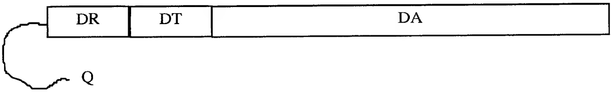
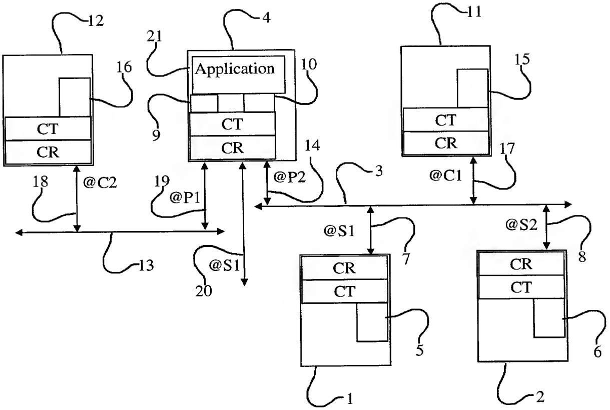

DR
DT
DA
Q

Fig.3


12
21
4
11
Application
10
16
15
9
CT
CR
CT
CR
CT
CR
14
17
3
18
@C2
19
@P1
@P2
@C1
@S1
@S2
7
8
13
@S1
20
CR
CT
CR
CT
5
6
1
2

Declaration and Power of Attorney For Patent Application
Declaration Pour Demandes de Brevets Avec Pouvoirs
French Language Declaration

En tant qu' inventeur nomme ci-après, Je déclare par le présent acte que:

Mon nom, mon domicile, mon adresse postale, ma nationalité sont ceux qui figurent ci-après,

Je déclare que je crois être l'inventeur original, premier et unique (si un seul nom figure sur le présent acte) ou un des co-inventeurs, originaux et premiers (si plusieurs noms figurent sur le present acte) du sujet revendiqué et pour liquel un brevet est demande sur la base de l'invention intitulée:

Relais d'accès à un réseau serveur,

transparent sur un réseau client

______________________________,

dont la description
(cocher la case correspondante)

☒ est annexée au présent acte.

☐ a été déposée ______________________

Numéro de série de la demande ______________________

et modifiée le ______________________.
(si approprié)

Je déclare par le présent acte avoir examiné et compris le contenu de la description identifiée ci-dessus, revendications y compris, et le cas échéant telle que modifiée par l'amendment cité plus haut.

Je reconnais le devoir de divulguer l'information qui est en rapport avec l'examen de cette demande selon Titre 37 du Code des Reglements Fédéraux §1.56(a).

As a below named inventor, I hereby declare that:

My residence, post office address and citizenship are as stated below next to my name,

I believe I am the original, first and sole inventor (if only one name is listed below) or an original, first and joint inventor (if plural names are listed below) of the subject matter which is claimed and for which a patent is sought on the invention entitled

____

____

______________________________,

the specification of which

(check one)

☐ is attached hereto.

☐ was filed on ______________________ as

Application Serial No. ______________________

and was amended on ______________________.
(if applicable)

I hereby state that I have reviewed and understand the contents of the above identified specification, including the claims, as amended by any amendment referred to above.

I acknowledge the duty to disclose information which is material to the examination of this application in accordance with Title 37, Code of Federal Regulations, §1.56(a).

French Language Declaration

Je revendique par le présent acte le bénéfice de priorité étrangère selon Titre 35, du Code des Etats-Unis, §119 de toute demande de brevet ou d'attestation d'inventeur énumérée ci-après, et j'ai identifié également ci-après toute demande étrangère de brevet ou d'attestation d'inventeur ayant une date de dépôt antérieure à celle de la demande pour laquelle la priorité est revendiquée.

I hereby claim foreign priority benefits under Title 35, United States Code, §119 of any foreign application(s) for patent or inventor's certificate listed below and have also identified below any foreign application for patent or inventor's certificate having a filing date before that of the application on which priority is claimed:

Prior foreign applications

Demande(s) de brevet anterieure(s) dans un autre pays:

Priority claimed

Droit de priorité revendiqué

FR 9911594	France	16 09 1999	☒	□
(Number) (Numéro)	(Country) (Pays)	(Day/Month/Year Filed) (Jour/Mois/Année de dépôt)	Yes Oui	No Non
			□	□
(Number) (Numero)	(Country) (Pays)	(Day/Month/Year Filed) (Jour/Mois/Année de dépôt)	Yes Oui	No Non
			□	□
(Number) (Numéro)	(Country) (Pays)	(Day/Month/Year Filed) (Jour/Mois/Année de depôt)	Yes Oui	No Non

Je revendique par le présent acte, le bénéfice selon Titre 35 du Code des Etats-Unis, §120 de toute(s) demande(s) américaines énumérée(s) ci-après et, dans la mesure où le sujet de chacune des revendications de cette demande n'est pas divulgué dans la demande américaine antérieure, de la façon définie par le premier paragraphe de Titre 35 du Code des Etats-Unis, §112, je reconnais le devoir de divulguer l'information pertinente selon Titre 37 du Code des Réglements Fédéraux, §1.56(a), toute information qui se présente entre la date de dépôt de la demande antérieure et la date de dépôt de la demande, soit nationale, soit internationale PCT.

I hereby claim the benefit under Title 35, United States Code, §120 of any United States application(s) listed below and, insofar as the subject matter of each of the claims of this application is not disclosed in the prior United States application in the manner provided by the first paragraph of Title 35, United States Code, §112, I acknowledge the duty to disclose material information as defined in Title 37, Code of Federal Regulations, §1.56(a) which occurred between the filing date of the prior application and the national or PCT international filing date of this application:

(Application Serial No.) (No. de Demande)	(Filing Date) (Date de Dépôt)	(Etat) (brevetée, pendante, abandonné)	(Status) (patented, pending, abandoned)
(Application Serial No.) (No. de Demande)	(Filing Date) (Date de Dépôt)	(Etat) (brevetée, pendante, abandonnée)	(Status) (patented, pending, abandoned)

Je déclare par le présent acte que toutes mes déclarations, à ma connaissance, sont vraies et que toutes les déclarations faites à partir de renseignements ou de suppositions, sont tenues pour être vraies; de plus, toutes ces declarations ont été faites en sachant que de fausses déclarations volontaires u autres actes de même nature sont sanctionées par une amende ou un empnsonnement, ou les deux, selon la Section 1001, du Titre 18 de Code des Etats-Unis et que de selles déclarations délibérément fausses peuvent compromettre la validité de la demande ou du brevet délivré.

I hereby declare that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true; and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under Section 1001 of Title 18 of the United States Code and that such willful false statements may jeopardize the validity of the application or any patent issued thereon.

French Language Declaration

POUVOIR: En tant qu'inventeur, je désigne l'(les) avocat(s) et/ou l'(les) agent(s) suivant(s) pour poursuivre la procédure de cette demande et traiter toute affaire la concernant supris du Bureau des Brevets et de Marques:

Harold L. Stowell, Reg. 17,233
Edward J. Kondracki, Reg. 20,604
Dennis P. Clarke, Reg. 22,549
William L. Feeney, Reg. 29,918
John C. Kerins, Reg. 32,421

5

POWER OF ATTORNEY: As a named inventor, I hereby appoint the following attorney(s) and/or agent(s) to prosecute this application and transact all business in the Patent and Trademark Office connected therewith. (*list name and registration number*)

Harold L. Stowell, Reg. 17,233
Edward J. Kondracki, Reg. 20,604
Dennis P. Clarke, Reg. 22,549
William L. Feeney, Reg. 29,918
John C. Kerins, Reg. 32,421

Adresser toure correspondance à:

Edward J. Kondracki, Esq.
KERKAM, STOWELL, KONDRACKI
& CLARKE, P.C.
5203 Leesburg Pike, Suite 600
Falls Church, VA 22041

Send Correspondence to:

Edward J. Kondracki, Esq.
KERKAM, STOWELL, KONDRACKI
& CLARKE, P.C.
5203 Leesburg Pike, Suite 600
Falls Church, VA 22041

Adresser toute communication téléphonique à: (Nom) (*Numéro de téléphone*)

Edward J. Kondracki, Esq.
(703) 998-3302

Direct Telephone Calls to: (*name and telephone number*)

Edward J. Kondracki, Esq.
(703) 998-3302

Nom complet du seul ou premier inventeur DUJONC Jean-Yves	Full name of sole or first inventor
Signature de l'inventeur Date 30 Septembre 1999	Inventor's signature Date
Domicile 27 bis avenue Pasteur 78580 Maule FRANCE	Residence
Nationalité Française FRX	Citizenship
Adresse Postale 27 bis avenue Pasteur 78580 Maule FRANCE	Post Office Address
Nom complet du second co-inventeur, le cas echeant MARTIN René	Full name of second joint inventor, if any
Signature de l'inventeur Date 30 septembre 1999	Second Inventor's signature Date
Domicile 32, rue Gometz 91440 Bures sur Yvette FRANCE	Residence
Nationalité Française FRX	Citizenship
Adresse Postale 32, rue Gometz 91440 Bures sur Yvette FRANCE	Post Office Address

(Fournir les mêmes renseignements et la signature de tout co-inventeur supplémentaire.)

(Supply similar information and signature for third and subsequent joint inventors.)

T2147-907163-US3782/JMD/PG(PCT)

UNITED STATES DESIGNATED/ELECTED OFFICE (D.O./E.O./US)

Applicant:	Jean-Yves DUJONC & Rene MARTIN
International Application No.:	PCT/FR 00/02469
International Filing Date:	7 September 2000
U.S. Serial No.:	
U.S. Filing Date:	May 16, 2001
For:	RELAY FOR ACCESSING A SERVER NETWORK, TRANPARENT TO A CLIENT NETWORK

CHANGE OF ADDRESS

Honorable Commissioner of Patents and Trademarks
Washington, D.C. 20231

Sir:

Effective immediately, please note our new correspondence address and telephone/fax numbers as follows:

Miles & Stockbridge P.C.
1751 Pinnacle Drive
Suite 500
McLean, VA 22102-3833
Telephone: 703-903-9000
Fax: 703-610-8686

Respectfully submitted,

MILES & STOCKBRIDGE P.C.

Date: May 16, 2001

By: Edward J Kondracki
Edward J. Kondracki
Registration No. 20,604

1751 Pinnacle Drive – Suite 500
McLean, VA 22102-3833
Tel.: 703/903-9000
Fax: 703/610-8686